



Universidad de las Ciencias Informáticas

Facultad 3

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

SACOI-AA

**Sistema para el análisis de cumplimiento de objetivos
instructivos utilizando técnicas de
Analíticas de Aprendizaje**

Autor: Yasmany Febles Espinosa

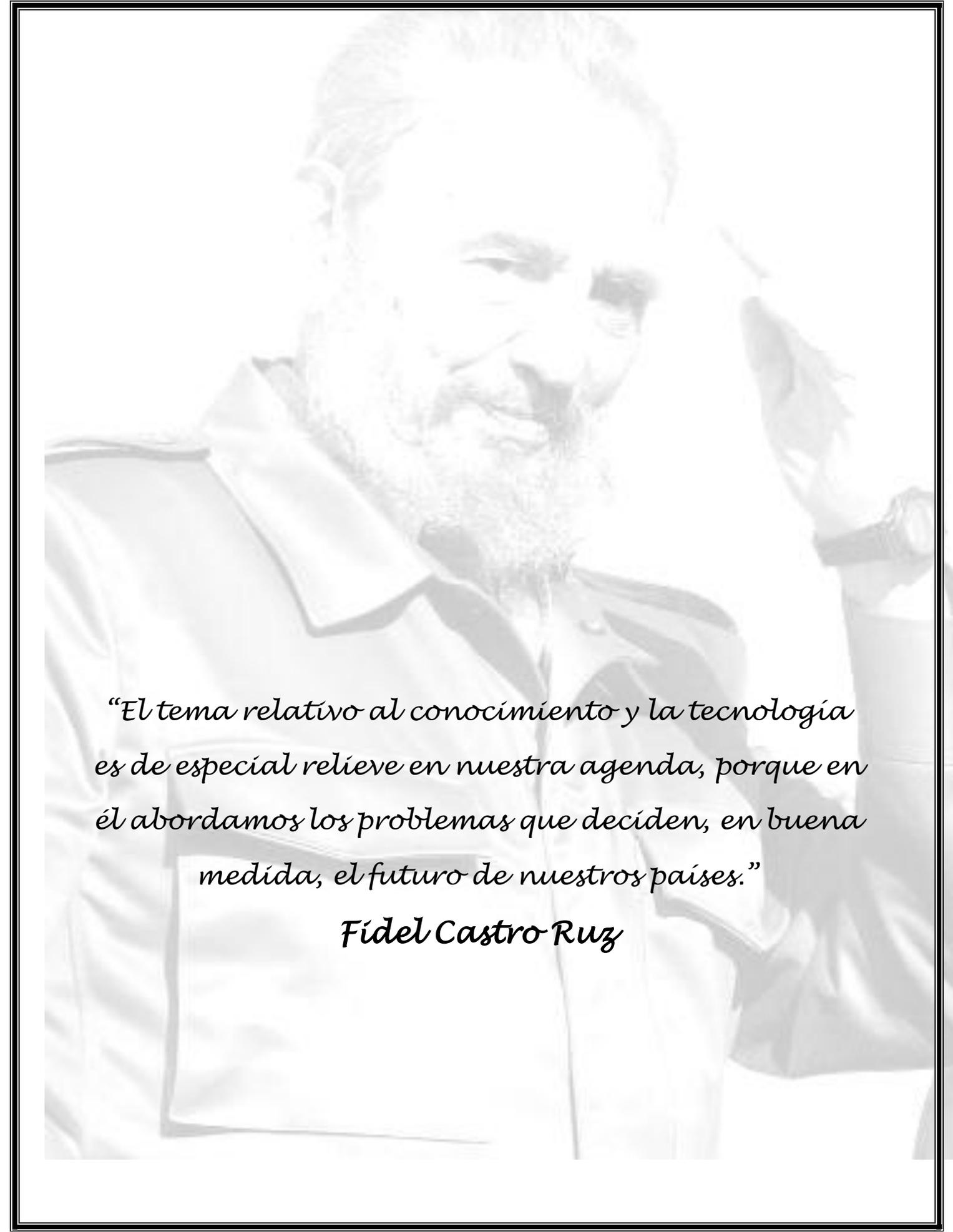
Tutor(es):

MSc. Naryana Linares Pons, Profesora Asistente

Dr. C. Edistio Yoel Verdecia Martínez, Profesor Titular

La Habana

Junio de 2013



“El tema relativo al conocimiento y la tecnología es de especial relieve en nuestra agenda, porque en él abordamos los problemas que deciden, en buena medida, el futuro de nuestros países.”

Fidel Castro Ruz

Declaración de autoría

Declaración de autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

(Autor)

(Tutor)

(Co-Tutor)

Datos de contacto

MSc. Naryana Linares Pons

Graduada de Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI), 2007, Ciudad de La Habana, Cuba; con Título de Oro. Profesora Asistente. Graduada de Máster en Gestión de Proyectos Informáticos, 2012.

Ha asistido a una decena de congresos y eventos nacionales e internacionales como delegado, ponente o autor. Tiene en su haber 20 publicaciones electrónicas en revistas y eventos. Ha cursado y aprobado veinticinco cursos de postgrado. Ha impartido docencia de pregrado desde el año 2007. Por sus resultados ha llegado a obtener el “Premio del Rector” de la Universidad de las Ciencias Informáticas 2011.

Edistio Yoel Verdecia Martínez

Graduado de Cibernética - Matemática en la Facultad de Matemática – Computación de la Universidad de la Habana (1995). Máster en Gestión de Proyectos Informáticos (2008) por la Universidad de las Ciencias Informáticas. Doctor en Ciencias Pedagógicas (2011). Profesor Titular.

Ha asistido a más de una veintena de congresos y eventos nacionales e internacionales. Consta en su haber más de 20 publicaciones electrónicas en revistas y eventos. Ha impartido docencia de pregrado y postgrado. Su actividad académico-investigativa se encuentra centrada en la aplicación de las TIC al proceso de enseñanza aprendizaje, sus resultados están respaldados por el impacto en la UCI y generalizada a otros centros del país. Ha impartido docencia y realizado misiones en otros países.

AGRADECIMIENTOS

Quisiera agradecer en primer lugar a la Revolución por formarme como un hombre de bien. A mis amigos de siempre que nunca olvidare lo que de ellos aprendí. A mis nuevos hermanos de distintos padres pero de una misma idea: Leiza por su apego, cariño y ayuda incondicional, a Cire por su locura pero siempre con la pizca de confianza y de ayuda, a Kirenia por ser mi persona de confianza. A mis verdaderos amigos Yennis y Josbel con los cuales siempre pude contar y por brindarme su apoyo en el momento necesitado.

A mis nuevos amigos que conocí por mi paso en la universidad, que sostuvieron mis manos cuando se me cansaron, a los que comenzaron la carrera y llegaron a la meta conmigo, en especial al Juanca, al Mega, Aroldo, Yessikita, a la Mary, a Rutmy y a Saly, a los muchachos del lab de proyecto, a mis compañeros de grupo que desde primer año vienen conmigo. A todos aquellos, que son muchos, los cuales no daría abasto esta página para poner y que contribuyeron a formar mi vida y la realización de este trabajo.

Agradezco de forma general a los que me ofrecieron su amistad y me brindaron su apoyo para el cumplimiento de este sueño, a Fidel por esta oportunidad de estudiar en una Universidad tan maravillosa, a todos aquellos que aunque no están presentes les hubiese gustado disfrutar de este inolvidable momento.

DEDICATORIA

DEDICATORIA

Quisiera dedicarle este título en primer lugar, a mis seres más queridos que siempre estuvieron apoyándome en el desempeño de mi carrera. A mi mamá que es el tesoro más grande que un hijo pudiera tener, por ser la guía y faro en mi carrera, por darme fuerza y seguridad para seguir adelante, por estar conmigo en los momentos más difíciles. A mis abuelos en especial a Mima y a mi abue Mario por confiar siempre en mí y hacerme ver que todo en la vida se logra con esfuerzo y dedicación, a mis tías especiales Tania e Idalmis por darme fuerza para seguir adelante en mi futuro, a mis hermanas Dayana y Yarisay que siempre fueron fuente de inspiración en mis estudios y a mi primo Osmany por ser tan especial conmigo y a todos los que siempre lucharon por verme felizmente graduado, a mi novia Liany que siempre estuvo a mi lado en estos casi 3 años, al igual que a su familia en especial Ida y de forma muy especial a la memoria de mi tía Villa que no pudo ver terminado mi sueño.

A mi tutora Naryana que a pesar de su corta edad fue para mí como una Madre, por sus oportunas sugerencias y por toda su ayuda, a mi tutor Edistio que siempre conté con su apoyo a pesar de la lejanía, a mi tribunal por siempre corregirme y ayudarme en el desarrollo de toda la tesis.

A los demás profesores por apoyarme en el desempeño de la tesis, por dedicarme parte de su tiempo y principalmente por ser tan especiales.

RESUMEN

La utilización de las Tecnologías de la Información y las Comunicaciones ha impactado favorablemente muchas áreas de la sociedad. Dentro de las áreas que ha recibido un impulso de ellas se encuentra la educación, desde la década del 90 del pasado siglo la introducción acelerada de las TIC en la educación ha sido un elemento a destacar. Las TIC han potenciado el desarrollo de nuevas técnicas para la creación de sistemas con el desarrollo en paralelo de la inteligencia artificial, la cual constituye actualmente un área de creciente interés dentro del análisis del proceso de enseñanza-aprendizaje. La creación de numerosos sistemas, apoyados en este campo de la Ciencia de la Computación aumenta considerablemente cada día en aras de mejorar el proceso. Una de las dificultades en las universidades actuales se centra en el aprovechamiento limitado de la información existente del proceso. En este trabajo se propone un sistema para minimizar el problema del limitado aprovechamiento de la información y los datos que se obtienen de los estudiantes durante el proceso de enseñanza-aprendizaje, lo cual dificulta el análisis del cumplimiento de los objetivos instructivos. Del estudio realizado sobre las técnicas de aprendizaje de inteligencia artificial, se explican sus características y conceptos fundamentales con el objetivo de lograr una propuesta sustentada por fundamentos teóricos de actualidad. El sistema fue probado desde el punto de vista informático, los resultados de este proceso son expuestos en la memoria de la investigación.

PALABRAS CLAVES: Inteligencia artificial, Sistemas tutores inteligentes, Sistemas adaptativos de enseñanza-aprendizaje.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	7
1.1 Breve caracterización sobre la utilización de las Tecnologías de la Información y las Comunicaciones en el proceso docente educativo	7
1.2 Caracterización de la Inteligencia Artificial y de su aplicación en el proceso de enseñanza-aprendizaje	10
1.3 Tendencias actuales en el desarrollo de la informática y su posible utilización en la educación	12
1.3.1 Computación en la nube	12
1.3.2 Analíticas de aprendizaje	14
1.3.3 Sistemas adaptativos inteligentes	16
1.4 Estudio de Técnicas de inteligencia artificial. Impacto y vigencia en la educación	17
1.4.1 Sistemas basados en Casos	18
1.4.2 Redes Neuronales	19
1.4.3 Sistemas basados en Reglas	21
1.4.4 Sistemas basados en probabilidades	22
1.5 Estudio de las técnicas y herramientas que soportan la solución	23
1.5.1 PHP	24
1.5.2 Java	24
1.5.3 PostgreSQL	25
1.5.4 MySQL	26
1.5.5 Zend Framework	27
1.5.6 Symfony	27
1.5.7 Extreme Programing	29
1.5.8 Rational Unified Process	30

1.5.9	Suite Visual Paradigm	32
1.6	Conclusiones parciales	32
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA		34
2.1	Propuesta de sistema	34
Fase de Exploración		34
2.2	Captura de los requisitos funcionales de software	34
2.3	Descripción de los requisitos no funcionales de software	37
2.4	Descripción de las historias de los usuarios	39
2.5	Estilo Arquitectónico	43
2.6	Patrón Arquitectónico.....	44
2.7	Patrones de Diseño	46
2.7.1	Patrones GRASP	46
2.7.2	Patrones GOF	48
2.8	Descripción de la arquitectura	49
2.9	Representación de los diagramas de Base de Datos	52
2.10	Conclusiones Parciales	54
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA		56
Fase Plan de lanzamiento.....		56
3.1	Plan de iteraciones.....	56
3.2	Plan de duración de las iteraciones.....	57
3.3	Plan de entregas.....	58
3.4	Representación del Diagrama de Despliegue	58
Fase Producción y Validación		59
3.5	Diseño y ejecución de las Pruebas de Software	59
3.6	Validación mediante pruebas unitarias	60

3.7 Validación mediante pruebas de aceptación.....	64
3.8 Validación mediante Técnica Delphi del criterio de expertos	67
3.9 Conclusiones Parciales	74
CONCLUSIONES GENERALES	75
RECOMENDACIONES	76
BIBLIOGRAFÍA	77

ÍNDICE DE FIGURAS

Figura 1. Representación de la estructura del conocimiento	23
Figura 2: Representación del patrón arquitectónico MVC.....	45
Figura 3: Representación del patrón MVC que implementa Symfony2.	45
Figura 4: Representación arquitectónica del sistema SACOI-AA en capas usando MVC.....	50
Figura 5: Representación del diagrama de clases.....	52
Figura 6: Representación del diagrama de Base de Datos.....	54
Figura 7: Representación del diagrama de despliegue.....	59
Figura 8: Representación del Método AdicionarRutaAction de la clase RutaController.....	62
Figura 9: Grafo de Flujo asociado al Método AdicionarRutaAction ()......	63
Figura 10: Rangos obtenidos a partir de los puntos de corte. Fuente: Elaboración propia.	72
Figura 11. Resultados de las encuestas de autoevaluación de expertos.	73
Figura 12: Categoría otorgada por los expertos al sistema SACOI-AA.	73

ÍNDICE DE TABLAS

Tabla 1: Descripción de la historia de usuario “Adicionar la ruta de aprendizaje de una asignatura”.	39
--	----

Tabla 2: Descripción de la historia de usuario “ <i>Modificar la ruta de aprendizaje de una asignatura</i> ”	40
Tabla 3: Descripción de la historia de usuario “ <i>Eliminar la ruta de aprendizaje de una asignatura</i> ”	40
Tabla 4: Descripción de la historia de usuario “ <i>Mostrar la ruta de aprendizaje de una asignatura.</i> ”	41
Tabla 5: Descripción de la historia de usuario “ <i>Crear evaluaciones que permitan comprobar el cumplimiento de los objetivos instructivos</i> ”	41
Tabla 6: Descripción de la historia de usuario “ <i>Graficar la ruta de aprendizaje de cada estudiante</i> ”	42
Tabla 7: Descripción de la historia de usuario “ <i>Emitir reporte del nivel de completamiento de los objetivos vencidos</i> ”	42
Tabla 8: Distribución de historias de usuario por iteración.....	58
Tabla 9: Caso de prueba de aceptación: HU10	65
Tabla 10: Caso de prueba de aceptación: HU11	65
Tabla 11: Caso de prueba de aceptación: HU12	66
Tabla 12: Caso de prueba de aceptación: HU13	66
Tabla 13: Caso de prueba de aceptación: HU14	67
Tabla 14: Cálculo del coeficiente de conocimiento.....	69
Tabla 15. Rango de los valores para el cálculo del coeficiente de Kendall. Fuente: Elaboración propia	71

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) son el resultado de las posibilidades creadas por la humanidad en torno a la digitalización de datos, productos, servicios, procesos y de su transportación a través de diferentes medios, a grandes distancias y en pequeños intervalos de tiempo, de forma confiable y con relaciones costo-beneficio nunca antes alcanzadas por el hombre (Castañeda Hevia 2003). Las tecnologías, como una de las expresiones más transformadoras alcanzadas por las ciencias técnicas, se componen de medios, procedimientos y recursos humanos preparados en el uso de esos medios y en el dominio de esos procedimientos y se diseñan, generalmente, a partir de las características del proceso dentro del cual van a ser utilizadas (Martínez Sánchez 2008).

La educación con la introducción de las TIC en cuanto a los modelos pedagógicos de enseñanza-aprendizaje utilizados dio un gran salto ya que fue un tránsito que se llevó a cabo desde la enseñanza tradicional en la que el papel del profesor era autoritario, impositivo y ejercía el poder en la transmisión del contenido, el cual a su vez era centrado en el conocimiento enciclopédico y el papel del estudiante era pasivo y receptor. En la actualidad esta enseñanza se encuentra asistida por las TIC y el papel del profesor ha cambiado pasando a ser mediador, guía, creador de condiciones favorables para el desarrollo de los conocimientos y a su vez el contenido es centrado en la destreza y en el tecnicismo que es crear habilidades en los estudiantes quienes tienen un papel activo, reflexivo, conscientes de su aprendizaje (Daniel 2005).

El modelo de enseñanza tradicional se ha ido transformando en paralelo con la introducción de las TIC y se ha visto reemplazado por modelos contruidos a partir de cambios que van ocurriendo y responden a necesidades de la sociedad que los motiva. Hay que ver las TIC como una oportunidad que permite la innovación hacia un modelo de enseñanza donde se conjugan entre estudiantes y profesores aspectos del desarrollo de la personalidad, el diálogo y la investigación de manera conjunta. Las TIC por sus características constituyen una oportunidad extraordinaria y al mismo tiempo un reto para la educación en todos los niveles.

El uso de las TIC en el proceso de enseñanza-aprendizaje ha tenido gran impacto dado que entre las ventajas que brindan se pueden mencionar: el trabajo en equipo, el acceso a información valiosa en las bibliotecas virtuales y la adquisición de conocimientos de forma motivadora, creativa e investigativa. Ha permitido la creación de la enseñanza en línea a través de nuevos entornos comunicativos como foros y blogs. Con el uso de las TIC además se

favorece el aprendizaje cooperativo al facilitar la organización de actividades grupales, brindando la posibilidad de adaptación de la información a las necesidades y características de los usuarios. Por otro lado se establecen nuevas formas de acceso a los grandes volúmenes de información que se generan durante el proceso de enseñanza-aprendizaje (PEA) y que suscita el uso de complejas bases de datos que garanticen los adecuados niveles de acceso mediante la organización de dicha información (Ferro Soto 2009).

De acuerdo con la revisión bibliográfica realizada, de donde se pueden citar a (Mesa Vázquez 2011) y (Ecurrred 2010), en Cuba el uso de las TIC se ha extendido a todos los niveles de enseñanza. El cambio de paradigma que se introdujo en la educación cubana fue grande ya que se transformó el papel de los profesores en cuanto a la forma de llevar a cabo el proceso de enseñanza-aprendizaje. Las TIC han propiciado la creación de espacios educativos virtuales basados en modelos pedagógicos que puedan garantizar el aprendizaje de los estudiantes utilizándose innovadoras estrategias. El desarrollo de las TIC en el país obedece a la voluntad política del gobierno que institucionalizó su empleo con la creación en el año 2000 del Ministerio de Informática y las Comunicaciones (MIC), con el objetivo de contar con un organismo que dirigiera el uso de las nuevas tecnologías. Dentro de las prioridades del MIC se encuentran revitalizar y actualizar bajo esta nueva óptica la estrategia de la informatización de la sociedad cubana.

En la actualidad Cuba se encuentra en un proceso de reestructuración de los organismos de la administración central, ello busca fortalecer el trabajo de los ministerios para que puedan cumplir la misión principal de dirigir, supervisar y controlar la política del Estado y el Gobierno. La reestructuración del MIC está orientada a lograr que el organismo esté a la altura del Sistema Único de Comunicaciones que requiere el país. Ahora cambia su denominación actual del Ministerio de Informática y las Comunicaciones por la de Ministerio de Comunicaciones. Desde ahora el organismo solo se encargará de las funciones estatales, mientras las empresariales serán asumidas por dos organizaciones superiores de dirección empresarial atendidas por el Ministerio (Gaceta Oficial 2013).

Para concretar la esta estrategia de acelerar la informatización del país y lograr la obtención de ingresos, se fundó en el año 2002 la Universidad de las Ciencias Informáticas (UCI). La UCI tiene como misión formar profesionales comprometidos con la Revolución y altamente calificados en la rama de la informática, con el objetivo de producir aplicaciones y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación desde la

producción con la finalidad de informatizar el país y desarrollar la industria cubana del software para contribuir al desarrollo económico. La UCI cuenta con laboratorios y aulas que disponen de alta tecnología para llevar a cabo el proceso de enseñanza-aprendizaje (Ríos Rodríguez 2008).

La universidad es una institución productiva que cuenta con una infraestructura tecnológica de apoyo a sus principales procesos en la que a partir de una red de área local se puede acceder a los principales sistemas que han sido implementados y que soportan la informatización de la misma. La plataforma de servicios con que cuenta la universidad tiene módulos que incluyen los procesos de ingreso, pregrado, investigación, producción, teleformación, extensión, residencia y biblioteca entre otros. Es importante hacer notar que existe un sistema para la Gestión Académica y un Entorno Virtual de Aprendizaje (EVA), soportado en la plataforma Moodle, donde los profesores pueden implementar estrategias para el proceso de enseñanza-aprendizaje complementarias a las clases presenciales, así como diseñar cursos semi-presenciales o totalmente a distancia, disponiendo los estudiantes de un medio en el cual pueden obtener, utilizar o compartir materiales didácticos.

La UCI también cuenta con un Centro de Innovación y Calidad de la Educación (CICE). Este centro tiene como misión contribuir a elevar la calidad de la formación en la UCI apoyando la superación permanente de los recursos humanos con el soporte de medios modernos de la informática y la tecnología educativa. En él se han desarrollado herramientas e instrumentos (herramienta Perfiles) para la captación de grandes volúmenes de datos con los que se pudieran trabajar de manera intencionada para mejorar el PEA.

De manera general en la formación de profesionales, los objetivos se estructuran para cumplir un encargo social, que es formar el profesional al que la sociedad aspira. En la educación superior cubana los objetivos son el elemento rector del proceso, se van derivando de los objetivos de la carrera, pasando por la disciplina y el año hasta llegar a la asignatura (Horruitiner Silva 2007). Cada asignatura tiene un determinado contenido, con sus correspondientes componentes (conocimientos, habilidades y valores), relacionados con las habilidades y los valores se encuentran los objetivos instructivos. Comprobar el nivel de desarrollo de los objetivos instructivos es un elemento importante a la hora de diseñar un PEA que se adapte a las características del grupo y de cada estudiante en particular, esta comprobación es necesaria para asignar la calificación a las diferentes evaluaciones, ya sean frecuentes, parciales o finales.

Sin embargo con el desarrollo de la investigación se determinó la siguiente situación problemática:

- Tendencia a la dispersión de los datos, que dificulta en gran medida la interoperabilidad de los sistemas de información, esto se hace más evidente en los sistemas que apoyan el PEA. Un ejemplo lo constituye en que el Sistema para la Gestión Académica y el EVA de la universidad no están conectados.
- Escasos mecanismos que posibiliten que los profesores puedan arribar a conclusiones sobre el estado en que se encuentra el desarrollo docente educativo a partir de los datos que procesan los sistemas.
- Insuficiente tratamiento y aprovechamiento de la información que aporta el estudiante durante el proceso docente educativo.
- Es difícil para muchos profesores jóvenes determinar el nivel de cumplimiento de los objetivos instructivos de la asignatura que imparten, pues en la mayoría de las ocasiones las tareas docentes aportan en el cumplimiento de más de un objetivo.

De ahí que se plantee el siguiente **problema a resolver**: el limitado aprovechamiento de la información y los datos que se obtienen de los estudiantes durante el proceso docente dificulta el análisis del cumplimiento de los objetivos instructivos.

Derivado del problema se plantea como **objeto de estudio**: las técnicas de inteligencia artificial en la implementación de sistemas para la educación que mejoren el actual aprovechamiento que se le da a la información.

Como **objetivo general** se propone: desarrollar un sistema para mejorar el aprovechamiento de la información y los datos que se generan de los estudiantes durante el proceso de enseñanza-aprendizaje a partir del análisis de cumplimiento de los objetivos instructivos.

Dentro de los **objetivos específicos** de la investigación se encuentran:

1. Formalizar el marco teórico conceptual de la investigación a partir del estudio de las principales técnicas de inteligencia artificial y específicamente de Analíticas de Aprendizaje en la Educación Superior.
2. Desarrollar un sistema para el análisis del cumplimiento de objetivos instructivos utilizando técnicas de Analíticas de Aprendizaje.
3. Validar la propuesta de solución mediante la utilización de técnicas científicas.

Queda delimitado como **campo de acción**: las Analíticas de Aprendizaje en el

perfeccionamiento de los procesos de enseñanza-aprendizaje.

A partir de todo lo cual se propone la siguiente **idea a defender**: el desarrollo de un sistema para mejorar el análisis de cumplimiento de objetivos instructivos contribuirá con un mayor aprovechamiento de la información y los datos que se gestionan de los estudiantes durante el proceso de enseñanza-aprendizaje.

Para la realización de la investigación se han utilizado los métodos científicos que se enuncian a continuación:

Como métodos teóricos:

- **Análítico-sintético:** Se utilizó para el desglose de la información en las diferentes áreas de importancia según los objetivos de la investigación, para facilitar su estudio, para después integrar esa información e ir construyendo el estado del arte y los otros aspectos básicos de esta investigación.
- **Análisis histórico-lógico:** Se utilizó para el estudio de la evolución y desarrollo histórico de la inteligencia artificial y de su aplicación en la enseñanza.
- **Inductivo-deductivo:** Empleado para abordar el objeto de estudio de la investigación hasta llegar a las conclusiones.

Como métodos empíricos:

- **Análisis documental:** para la elaboración del marco teórico de la presente investigación y la argumentación de la situación actual.
- **Encuesta:** Se realizó para obtener información necesaria a partir de las respuestas de varios especialistas sobre el trabajo con las técnicas de IA.
- **Consulta de expertos:** Se utilizó para obtener el criterio de los especialistas para guiar la investigación por el camino correcto.

Para una mejor comprensión del documento se propone la siguiente organización: resumen, introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos.

En el Capítulo 1: Se realiza un estudio donde se investiga acerca del uso que tienen las TIC en la educación, se realiza una profundización de las tendencias en el desarrollo de la informática, así como de la aplicación de la inteligencia artificial en la enseñanza, que dan paso a la creación de los sistemas expertos, también se hace una disertación de las técnicas y herramientas que se utilizan para la creación de los sistemas adaptativos en la enseñanza-

aprendizaje.

En el Capítulo 2: Se describen las principales características del sistema a desarrollar. Así como la especificación de los requisitos funcionales que son las funcionalidades que brindará la aplicación y los requisitos no funcionales que serían las prestaciones que deben presentar tanto los servidores para brindar el buen funcionamiento del sistema como las estaciones clientes para hacer uso de las mismas. Además se describen los principales artefactos que genera la metodología de desarrollo XP.

En el Capítulo 3: En este capítulo se lleva a cabo el desarrollo de la implementación y validación de la propuesta del sistema. Se realiza el diagrama de despliegue correspondiente para llevar a cabo una buena implantación del sistema solicitado, así como se desarrolla un modelo de prueba para comprobar la efectividad del sistema.

Finalmente se establecen conclusiones, recomendaciones y anexos que permiten completar la memoria escrita de la tesis.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Este capítulo tiene como objetivo presentar un estudio sobre el impacto que ha tenido en la educación la introducción de las tecnologías de la información y las comunicaciones (TIC). Además se abordan las tendencias que existen en la actualidad en el desarrollo de la informática como Computación en la Nube, las Analíticas de Aprendizaje y los Sistemas Adaptativos de Enseñanza-Aprendizaje, así como su evolución e impacto en procesos docentes. Finalmente se ofrecen conclusiones parciales del capítulo.

1.1 Breve caracterización sobre la utilización de las Tecnologías de la Información y las Comunicaciones en el proceso docente educativo

Las Tecnologías de la Información y las Comunicaciones tienen inmensas repercusiones en prácticamente todos los aspectos de la vida. Estas tienen la capacidad de reducir el tiempo y la distancia, posibilitando su uso en beneficio de millones de personas en todo el mundo. Con la aparición de las TIC se experimentaron grandes cambios en diversos sectores de la sociedad entre ellos el sector de la educación donde por sus características el desarrollo de las tecnologías constituyen una oportunidad extraordinaria y al mismo tiempo un reto para todos los involucrados en el proceso de enseñanza-aprendizaje. Es así como se han desarrollado nuevas formas de enseñar y de aprender (Castañeda Hevia 2003).

A juicio del autor, las Tecnologías de la Información y las Comunicaciones tienen un gran potencial para favorecer el progreso de los alumnos y de los profesores, pero solo si son utilizadas de forma apropiada. La realidad que se está viendo es que las tecnologías de la información están transformando y engrandeciendo las experiencias de las nuevas generaciones (C. Berliner 2006). Las ventajas que se han atribuido a las TIC como instrumentos de mejora del aprendizaje de los alumnos son numerosas. Entre estas se pueden encontrar: la capacidad para crear contextos de aprendizaje que abren nuevas posibilidades de información y de comunicación, la interactividad que posibilita que los estudiantes pueden adentrarse con más facilidades en experiencias de aprendizaje en las que reciben nueva información, y por último que los programas informáticos pueden transformar informaciones abstractas en modelos figurativos, lo que facilita su comprensión y su aprendizaje (C. Berliner 2006).

La universidad no es más que una instancia superior en la educación. Su modelo gira alrededor

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

de tres procesos básicos o sustantivos: docencia, investigación y extensión. Además como sucede en otras empresas, su modelo de funcionamiento se encuentra muy ligado a las TIC. Sobre el uso de las tecnologías de la información y las comunicaciones en la educación superior hay muchas expectativas en cuanto a la mejora de los procesos que se ejecutan en cada uno de sus ejes principales.

Las TIC tienen tres objetivos fundamentales en la educación superior: ser un medio que facilite el desarrollo del proceso enseñanza-aprendizaje, incrementar la competitividad de las universidades y facilitar la movilidad mediante el desarrollo de portafolios digitales para los involucrados. Las TIC pueden resultar decisivas para potenciar de forma considerable que se logren las distintas misiones que traen aparejadas las universidades cuando son creadas: ser una universidad-empresa, parques científicos y tecnológicos, regiones de excelencia innovadora, prestación de servicios técnicos y de asesoría, incubadoras de empresas, centros de formación continua, foros de debate, observatorios de estudios sociales, oficinas de cooperación, entre otras. Por lo tanto, las universidades deberán dotarse de las TIC para ser globalmente competitivas y localmente comprometidas (C. Berliner 2006).

Una universidad como institución está obligada a contar con las infraestructuras y procesos tecnológicos que sean necesarios para cumplir con el desarrollo de sus ejes fundamentales. En el caso de la docencia estas instituciones tendrán que potenciar sus plataformas tecnológicas con las TIC como herramientas de trabajo y de apoyo. Por lo que respecta a la infraestructura de redes y comunicaciones, la Web ha sido uno de los primeros servicios que se han ofrecido a través de internet y, sin duda, el más extendido. Servicios como la transferencia de ficheros, el correo electrónico y la videoconferencia forman parte, cada vez con más auge de la vida universitaria.

Actualmente los grandes adelantos se encuentran marcados por el fenómeno de la «Web 2.0» cuya principal característica consiste en el surgimiento incesante de nuevas aplicaciones web, que están cambiando la manera en que el usuario hace uso de la red, potenciando aspectos como la creatividad, compartición de información y la colaboración. Así, se desarrollan comunidades como son las llamadas redes sociales, wikis y blogs para las cuales solo se necesita tener un cable de conexión. Los contenidos pueden ser combinados y personalizados de diferentes formas (Esteve 2009).

Desde la perspectiva de la docencia universitaria en el espacio europeo de educación superior (EEES), las TIC vistas como un medio para conseguir el objetivo del aprendizaje y no como el

fin de las estrategias de gestión universitaria, tienen tres objetivos fundamentales. El primer objetivo es el de ser un medio que facilite el desarrollo del proceso de enseñanza-aprendizaje como apoyo a la docencia presencial, deben servir de base para el establecimiento de procesos formativos en línea y, sobre todo, han de facilitar la acción tutorial (C. Berliner 2006; Esteve 2009).

Una breve caracterización del proceso de formación lleva a expresar que la formación de profesionales en las instituciones de la educación superior se centra alrededor del modelo del profesional, modelo que expone la aspiración de la sociedad. Esta aspiración se organiza en la carrera, partiendo de la definición de los problemas que el profesional debe resolver, este proceso de formación es una actividad compleja, en la que se delimitan los contenidos a impartir a los estudiantes y como estos se van desagregando hasta llegar a las asignaturas, donde el objetivo juega un papel esencial.

El objetivo, definido desde la didáctica, es lo que queremos lograr en el estudiante, son los propósitos y aspiraciones que pretendemos formar en los alumnos. El objetivo es el componente rector del proceso docente-educativo (de enseñanza-aprendizaje), o lo que es lo mismo, el más importante, porque es él, el que en un lenguaje pedagógico, explícita la solución de los problemas, de la necesidad social. Al interpretar el proceso docente educativo como actividad orientada al cumplimiento de los objetivos se presupone que el mismo se desarrolla a través de una sucesión de tareas docentes (Álvarez Zayas 1996).

Los objetivos se clasifican, de acuerdo con el grado de trascendencia en la transformación que se espera alcanzar en los estudiantes, en instructivos y educativos. El instructivo se refiere a las transformaciones que en el pensamiento queremos alcanzar en los estudiantes; y el educativo, en las transformaciones a lograr en los sentimientos, convicciones y otros rasgos de la personalidad de los escolares (Portuondo 2001).

Según la definición anterior la instrucción se concreta, como se refiere al pensamiento, en las capacidades y habilidades a formar, las que siempre están asociadas a un conjunto de conocimientos. De tal modo, al redactar el objetivo instructivo se debe, ante todo, precisar la habilidad que debe mostrar el estudiante si ha logrado el objetivo. Esta es el núcleo del objetivo instructivo.

Es por ello que todo proceso se organiza para que cada asignatura aporte en la formación del profesional que la sociedad demanda. En este proceso, la secuencia de tareas y su evaluación va aportando elementos que permiten al final del proceso definir el grado de cumplimiento de

los objetivos instructivos y educativos. En la práctica este es un proceso complejo y la utilización de las TIC puede aportar elementos que faciliten este proceso a los profesores y estudiantes, pero la realidad a partir de lo analizado, es que son pocos los sistemas que aportan elementos sobre el cumplimiento de estos objetivos. Esta es una de las dificultades de las herramientas informáticas que hoy se utilizan en las universidades, incluyendo dentro de ellas los entornos virtuales de aprendizaje.

Para los propósitos de esta investigación se asume que tal como plantea en la bibliografía consultada (Portuondo 2001), *“el objetivo instructivo, expresado en términos del aprendizaje, del resultado que se espera alcanzar en el estudiante, se puede plantear en lenguaje de acciones”*, se asume que estas acciones están vinculadas a las tareas docentes cuando el proceso se desarrolla en el marco del aula. Por tanto se puede realizar el diseño instruccional del proceso partiendo de uno o varios objetivos instructivos a cumplir y descomponiendo estos en tareas docentes que se interrelacionan y/o complementan. A este diseño de una secuencia de tareas docentes, interrelacionadas y/o complementadas, que tiene el fin de cumplir con uno o varios objetivos instructivos se le denominará *“ruta de aprendizaje”*. Es válido destacar que pueden existir múltiples rutas de aprendizaje para lograr un mismo grupo de objetivos instructivos.

Como último elemento es necesario destacar que aunque la presente investigación se centra en el cumplimiento de los objetivos instructivos no se pueden olvidar los objetivos educativos, a partir de considerar la unidad entre la instrucción y la educación. A pesar de esta estrecha interrelación determinar el cumplimiento de estos objetivos educativos es un proceso complejo, pues ellos al decir de (Álvarez Zayas 1996) y (Portuondo 2001), tienen como elemento que los constituye a los hábitos, las capacidades y las convicciones, elementos que no son fáciles de evaluar; es por ello que la investigación se centra en los objetivos instructivos esencialmente.

1.2 Caracterización de la Inteligencia Artificial y de su aplicación en el proceso de enseñanza-aprendizaje

El término inteligencia artificial (IA) fue sugerido por J. McCarthy, profesor de la universidad de Stanford en 1956 y se usa principalmente para referirse a una parte de la informática dedicada al diseño de máquinas que sean capaces de simular algunas de las conductas realizadas por el ser humano y que habitualmente se catalogan *“inteligentes”* (Gross Salvat 1992).

Cinco son las áreas más activas de investigación en la IA: el lenguaje natural, la robótica, las interfaces de comunicación, la programación automática y los sistemas expertos. Esta última

área es la que tiene en la actualidad una clara conexión con el desarrollo del software educativo. El término sistema experto no describe un producto sino más bien un conjunto de conceptos, procedimientos y técnicas que permiten utilizar la informática en una nueva dimensión (Gross Salvat 1992).

Los sistemas expertos permiten ayudar en el análisis y resolución de problemas complejos. Intentan simular la conducta de un experto humano con un dominio específico de conocimientos en un tema. Debe contener todo el contenido que manejaría el experto y la forma de razonamiento utilizada por éste, es decir, la manera en que el experto utiliza el conocimiento.

Con este tipo de sistemas se extiende el tipo de utilización de los ordenadores actuales ya que éstos no se centran únicamente en la realización de labores primordialmente repetitivas sino que se pasa a una utilización del ordenador como herramienta de ayuda en tareas complejas tales como el diagnóstico o la toma de decisiones (Gross Salvat 1992).

La aplicación de la inteligencia artificial en la educación, constituye actualmente un campo de creciente interés articuladas al desarrollo de sistemas de enseñanza asistida por ordenador (EAO) con el propósito de construir sistemas de enseñanza inteligentes. Las aplicaciones educativas a partir del uso de técnicas de IA varían básicamente, en función de los objetivos que se persiguen con el diseño del programa así como del tipo de conocimiento que la aplicación debe manipular (Soledad González 2004) (Barrón Estrada 2011).

Entre las aplicaciones que permiten crear programas destinados a la enseñanza se pueden distinguir dos grandes áreas: los micromundos y los tutores inteligentes (TI). Ambas modalidades se encuentran enraizadas en dos posiciones diferentes de la utilización de la informática en el terreno educativo (Barrón Estrada 2011) (Soledad González 2004).

La construcción de micromundos se sitúa en la búsqueda de herramientas que ayuden al sujeto a construir el conocimiento a partir de la interacción con el ordenador mediante procesos de aprendizaje basados fundamentalmente en el descubrimiento. Mientras los TI al igual que el software propio de la enseñanza asistida por ordenador tienen como objetivo la creación de programas que permitan transmitir un conocimiento previamente formalizado adoptando un tipo de conducta más «inteligente» que mejore el proceso de enseñanza a través de la máquina (Soledad González 2004) (Barrón Estrada 2011).

El autor es del criterio que con los tutores inteligentes se trata de crear un programa de instrucción y de interacción que pueda parecerse, lo más posible, a un profesor en su actividad

de enseñanza. Resaltando su capacidad para detectar las lagunas de conocimiento del alumno, acercarse a cada situación de aprendizaje y utilizar estrategias tutoriales a la medida del estudiante.

A los programas de EAO se les critica fundamentalmente su incapacidad por mantener un diálogo abierto con el alumno. Siempre es el ordenador el que presenta la información y espera la respuesta del alumno sin poder éste adquirir otra información en cualquier momento de la interacción. Además del aspecto comunicativo, el software desarrollado sigue estructuras de transmisión de conocimiento de carácter secuencial previamente determinadas y, por consiguiente, los programas carecen de la capacidad de una adaptación real a las necesidades y al estilo de aprendizaje individual del alumno (Barrón Estrada 2011) (Soledad González 2004).

Los tutores inteligentes tienen las ventajas de poder adaptarse a las características y ritmo de aprendizaje de cada alumno y proporcionar una ayuda también adaptable, la cual es capaz de ofrecer a los alumnos elementos de auto reflexión sobre su propio rendimiento pudiendo incluso compararse con otros estudiantes, reales o modelados. Con los tutores inteligentes se está mucho más interesado en ir evaluando las diferentes respuestas dadas por el alumno a lo largo de su interacción con el ordenador para detectar aspectos tales como el estilo del aprendizaje del alumno, su grado de interés y comprensión (Soledad González 2004) (Barrón Estrada 2011).

1.3 Tendencias actuales en el desarrollo de la informática y su posible utilización en la educación

A continuación se realiza el análisis de tres de las tendencias actuales en el desarrollo de la informática: la computación en la nube, las analíticas de aprendizaje y los sistemas adaptativos inteligentes; y se analiza además su utilización en el campo de la educación.

1.3.1 Computación en la nube

Los recientes avances en programación y tecnologías de información (TI), han mostrado también nuevas aplicaciones para Internet. En muchos entornos se habla hoy día de computación en la nube (Cloud Computing (CC) por su nombre en inglés). Aunque el término puede verse con diferentes significados, la definición más difundida se refiere a una situación en la cual la computación se hace en un sitio remoto, en lugar de hacerlo en una computadora de escritorio o en una portátil, usando para ello Internet. La nube es un sistema computacional

inteligente, complejo y poderoso al cual los usuarios simplemente se conectan (Dikaiakos 2009) (Ávila Mejía 2011).

Las operaciones computacionales a distancia, incluyendo el almacenamiento, el procesamiento y la distribución de información y comunicación empezaron a generalizarse en la última década del siglo pasado en entornos empresariales y corporativos. Sin embargo, su uso se ha extendido como un servicio fundamental en las aplicaciones web más recientes. Aunque este concepto no es nuevo ya que lleva más de veinte años de aplicación, está cambiando por su reciente generalización en las prácticas cotidianas de los usuarios en el uso de los ordenadores y el uso social que se hace de la información y de la comunicación transformada digitalmente por ellos (Mell 2011).

A partir de lo anterior se percibe que el concepto computación en la nube, denomina en la actualidad a una de las operaciones técnicas y comunicativas de la red. La “computación en la nube” no es más que mover o trasladar la información a través de la red, de forma aparentemente deslocalizada. Esta información puede ser trasladada en forma de datos, textos, fotografías, vídeos o también de aplicaciones, de partes de aplicaciones o incluso un sistema operativo completo, de una organización o del ordenador personal individualizado, a una ubicación remota (Johnson 2011).

Ese emplazamiento remoto físicamente para el usuario, habitualmente está compuesto por un grupo de servidores. Es así donde es almacenada la mayor parte de los datos y de las operaciones que con estos se realiza, es de estos servidores remotos que se recibe la prestación del servicio solicitado por los ordenadores locales. Se usa el término “nube” para hacer referencia a la flexibilidad pues da a entender que el servicio puede tomar formas muy variadas. La nube es un sistema computacional al cual la gente simplemente se conecta y para ello solo necesita un cable de red, independiente del dispositivo o cliente que utilice para ello (Ruiz del Olmo 2010).

La compañía Sun Microsystems hacía referencia a finales de la pasada década a que la propia red se convertiría en el ordenador, transformando de alguna forma el concepto que de este se tenía. Literalmente entonces “computación en la nube” es como se señala en la literatura consultada una transformación de los recursos informáticos, como los propios ordenadores y todo lo que contienen incluyendo los programas y sobre todo el almacenamiento de información, en un servicio web adicional (Cusumano 2010).

Sobre este tema algunos de los investigadores (Johnson 2011) (Cusumano 2010) plantean que

se trata de una forma de acceso, modificación y de intercambio que en muchos aspectos resulta análoga a servicios de compañías tradicionales como la electricidad o el agua. Además plantean como beneficios de estas tecnologías el ahorro cuantioso simplemente en el espacio físico necesario para los dispositivos, en los costes de mantenimiento, en la inversión en software y el hardware.

Los servicios que son brindados a través del uso de esta tecnología resultan gratuitos o de pago bajo demanda. El autor se refiere a la utilización de herramientas informáticas y de aplicaciones que no se encuentran físicamente en el dispositivo local pero que son ejecutadas en él a través de la red. Esta tecnología permite que los trabajos se hagan en lugares remotos y no en el ordenador personal usando para el movimiento de los servicios de aplicación a la red de redes "internet".

El CC refuerza la tendencia a la virtualización a todos los niveles de enseñanza y en todos los países. El desarrollo de las TIC hace posible que los nuevos estilos de educación a distancia cobren auge e implementen técnicas de enseñanza-aprendizaje en las que se tengan como centro del proceso al sujeto y a las tecnologías. El CC como resultado de la implementación continua de las tecnologías hace posible en el campo de la educación y específicamente en la enseñanza universitaria que se fortalezcan el auto aprendizaje, la auto gestión del conocimiento y de manera general se logre una mayor independencia de los estudiantes durante el proceso de enseñanza-aprendizaje.

1.3.2 Analíticas de aprendizaje

Las analíticas de aprendizaje se refieren a la interpretación de un amplio rango de datos producidos y recogidos de los estudiantes para orientar sobre su progresión académica, predecir actuaciones futuras e identificar elementos problemáticos en su aprendizaje. Los datos se recogen a partir de acciones explícitas, como completar tareas y realizar exámenes pero también de las actuaciones tácitas y he aquí lo novedoso, son interacciones sociales online, actividades extracurriculares y posteo en foros. La analítica de aprendizaje es la medición, recolección, análisis y reporte de datos acerca de los aprendices y sus contextos, con el objetivo de comprender, optimizar el aprendizaje y el entorno en que ocurre (J. Goldstein 2005) (Johnson 2011).

Las técnicas de analíticas de aprendizaje no son nuevas ya que los exámenes, la evaluación, la investigación institucional y la retención fueron medios de llegar a conclusiones sobre el desempeño de los estudiantes por parte de los docentes. Ahora en cambio la cantidad de

datos, modelos, personalización, adaptatividad, técnicas, fuentes de datos y poder de procesamiento de estos datos son los nuevos medios de apoyo a la toma de decisiones cuya actualidad cobra auge en la medida en que se acrecienta el desarrollo de las TIC. Las analíticas de aprendizaje son técnicas sofisticadas que se utilizan para mejorar el aprendizaje y la educación.

El objetivo de las analíticas de aprendizaje es permitir a docentes y centros universitarios adaptar las oportunidades educativas al nivel de necesidad y capacidad de cada estudiante. También pueden ser utilizadas para evaluar planes de estudio, programas e instituciones. A través de sus técnicas ayudan a los profesores, a los alumnos y a el centro que las esté utilizando a tomar decisiones para desarrollar nuevos planes (Johnson 2012). Estas técnicas ofrecen información en tiempo real lo que permite hacer modificaciones o ajustes inmediatos y así configurar un modelo más flexible y abierto al cambio. La analítica de aprendizaje puede ser usada para hacer perfiles e incluso predecir a los estudiantes que pueden estar en riesgo, analizando datos demográficos y de desempeño de otros estudiantes. Es una ayuda para la evaluación diagnóstica y formativa, para la adaptabilidad de los recursos de aprendizaje y para mejorar la planificación personalizada (Johnson 2011).

Las analíticas de aprendizaje bien utilizadas ayudarán a identificar las necesidades que los estudiantes tienen con su aprendizaje. Conducirán de manera eficiente la formación del conocimiento en los estudiantes. Estas técnicas ayudan a saber cómo ven los profesores el proceso de enseñanza-aprendizaje y la evaluación. Estas técnicas también enfrentan algunos desafíos, ya que requieren la combinación de datos procedentes de diferentes fuentes, a menudo en diversos formatos. También debe hacer frente al aspecto de la privacidad y del perfil del estudiante, que está siendo reducido a números e información. Los docentes necesitarán comprender tanto su potencial técnico como su utilidad pedagógica (Johnson 2011).

En la educación superior tienen grandes usos para tomar decisiones acerca de las admisiones, el aprendizaje y la retención de los estudiantes. En la era de los grandes datos, las analíticas de aprendizaje hacen mucho más que reportar. Al aumentar el volumen, la variedad, variabilidad y la velocidad del procesamiento se pueden hacer las siguientes preguntas: ¿Qué pasó?, ¿Con cuánta frecuencia y dónde?, ¿Cuál es exactamente el problema?, ¿Qué acciones se necesitan?, ¿Por qué está sucediendo?, ¿Qué sucede si las tendencias continúan?, ¿Qué sucederá a continuación?, ¿Qué es lo mejor que puede pasar? (J. Goldstein 2005).

La analítica de aprendizaje le añade valor a las bases de datos de las universidades ya que la información derivada de las mismas se puede convertir en un punto estratégico y de ventaja competitiva para la toma de decisiones docentes. Recientemente, el interés en cómo estos datos se pueden utilizar para mejorar la enseñanza y el aprendizaje ha visto crecimiento sin precedentes. El principal objetivo es observar y comprender los comportamientos de aprendizaje con el fin de permitir una intervención adecuada de los docentes (Daniel 2005).

1.3.3 Sistemas adaptativos inteligentes

Con el permanente y acelerado desarrollo de las TIC, es muy común escuchar hablar de entornos virtuales de aprendizaje, de ambientes virtuales de enseñanza y ambientes de enseñanza-aprendizaje asistidos por computadora (AEAC) (Ríos Rodríguez 2009), los cuales son el resultado de hibridar la evolución de la enseñanza mediante la utilización de las computadoras, la introducción de las Tecnologías de la Información y las Comunicaciones en la educación y del empleo de la inteligencia artificial. Los ambientes virtuales de enseñanza constituyen un sistema de herramientas para la organización de contenidos, ejemplos y conceptos en la toma de decisiones acerca de cómo mejorar el aprendizaje de un alumno y el desarrollo de su proceso de enseñanza-aprendizaje.

Una de las ventajas de los sistemas adaptativos de aprendizaje es su capacidad de adaptarse a las habilidades, conocimientos y características personales de un estudiante en particular. Esta adaptabilidad se puede lograr a través del modelado de las propiedades del estudiante. La presentación adaptativa de estos sistemas se centra en mostrar la información de diferentes maneras y reordenarla según el usuario que lo esté utilizando.

Los señalamientos en los reportes del autor (Johnson 2012) plantean que los sistemas adaptativos de aprendizaje son vistos como el siguiente paso lógico en el proceso continuo que comienza con los entornos personales de aprendizaje. Es la vinculación de datos de análisis de aprendizaje con el uso del software para modificar el ambiente de aprendizaje según sea necesario. Los sistemas adaptativos de aprendizaje son concebidos para permitir que las herramientas, los contenidos, ejemplos y conceptos puedan ser modificados en tiempo real basándose en cómo los estudiantes están aprendiendo realmente.

Los sistemas adaptativos de aprendizaje tienen gran relevancia para la docencia ya que pueden atender a los estudiantes con diferentes estilos de aprendizaje. Permiten que las modificaciones de los ambientes de aprendizaje sea en función de cómo cada estudiante está trabajando, atendiendo a sus necesidades específicas en el aprendizaje. Muestra como estar

más a tono con las necesidades de los alumnos implicados en el proceso. Los educadores pueden utilizar los datos para desarrollar nuevos cursos y nuevos materiales que se adapten a los diferentes estilos de aprendizaje (Martinez Sanchez 2008).

La mayoría de los investigadores (Ríos Rodríguez 2009) (Johnson 2012) plantean que los sistemas adaptativos de aprendizaje son programas que portan conocimientos de cierto contenido mediante un proceso interactivo. Proporcionan evaluaciones en tiempo real a los estudiantes, para que estos puedan concentrarse más en las habilidades que necesitan dominar. Tienen la idea de que una mayor investigación en datos de los estudiantes y el análisis de su aprendizaje permitirán a los profesores tener mayores conocimientos de los progresos del aprendizaje de cada uno de sus estudiantes, e intervenir en tiempo real.

Los sistemas adaptativos inteligentes utilizan una estrategia de navegación adaptativa. Esta estrategia permite la modificación instantánea de las secciones de contenidos a visitar por los estudiantes en correspondencia con su evaluación de desempeño. Así los contenidos presentarán diferente estructura de navegación de acuerdo al tipo de estudiante que los utilice. Estos ajustes están diseñados para ayudar a los estudiantes de forma individual, llegando al cumplimiento de sus habilidades y con la intención de elevar sus conocimientos (Johnson 2012).

Luego del estudio realizado de las principales tendencias en el desarrollo de la informática, de ver sus ventajas y limitantes, así como su aporte al desarrollo de un sistema adaptativo para el proceso de enseñanza-aprendizaje, el autor decide guiar el desarrollo del sistema con la utilización de las técnicas de analíticas de aprendizaje. Estas técnicas ayudan a realizar un análisis de los datos acerca de los estudiantes. El principal objetivo es permitirle a los docentes y a los centros universitarios adaptar las oportunidades educativas al nivel de necesidad y capacidad de cada estudiante.

1.4 Estudio de Técnicas de inteligencia artificial. Impacto y vigencia en la educación

Con el desarrollo de aplicaciones de inteligencia artificial es necesario hacer uso de distintas técnicas de aprendizaje para lograr que los sistemas informáticos sean capaces de sintetizar el conocimiento necesario con el objetivo de desarrollar una tarea sin necesidad de interacción humana. Las técnicas de IA según refiere la literatura consultada son promotoras de soluciones basadas en la computación. A su vez han mostrado beneficios reales al ser aplicadas en diversos sectores aportando ventajas competitivas, ya sea ahorrando tiempo o dinero, aumentando la eficiencia de algún proceso o aprovechando oportunidades aún no explotadas

(Martínez Labaut 2008).

Una de las técnicas de la IA más utilizada en la educación y que han marcado los grandes adelantos que en esta rama se han obtenido son los sistemas expertos. Existen diferentes tipos de sistemas expertos y cada uno de ellos tiene sus utilidades en la vida real. Estos sistemas son clasificados en dependencia de la forma que utilizan el conocimiento y de la forma que utilizan para representarlo. A continuación se ofrece un estudio de las formas de representación del conocimiento de los sistemas expertos que con mayor fuerza impactan en los sistemas para la educación.

1.4.1 Sistemas basados en Casos

El Razonamiento Basado en Casos (conocido por sus siglas como RBC) es una de las especificaciones de los sistemas expertos que se preocupa por el estudio de los mecanismos mentales necesarios para repetir lo que se ha hecho. Un Sistema Basado en Casos (SBC por sus siglas) forma parte del conjunto de sistemas basados en conocimiento. Ellos apoyan sus resultados en ejemplos que son los (casos) que se almacenan en la fase de aprendizaje. A través de una función de semejanza se determinan los casos más semejantes al nuevo problema y las soluciones de los casos recuperados se adaptan para obtener una solución. Durante este proceso se presentan dos problemas fundamentales: lograr una representación de la memoria de casos que permita una recuperación eficiente y considerar la incertidumbre presente en el conocimiento para mejorar la eficiencia en el proceso de adaptación (Martínez Labaut 2008).

El RBC se utiliza en todos los métodos de clasificación que permiten resolver problemas nuevos a partir de experiencias viejas. Su entrenamiento es simplemente almacenar los casos anteriores y no necesitan crear reglas. El tipo de aprendizaje que utilizan estos algoritmos es conocido como perezoso (Arencibia Ramirez 2011).

Estos sistemas basados en conocimiento se caracterizan por el hecho de utilizar directamente la información almacenada en la memoria del sistema sobre los problemas o casos ya resueltos o no. El RBC representa un nuevo método para resolver problemas no estructurados, en el cual el razonamiento se realiza a partir de una memoria asociativa que usa un algoritmo para determinar una medida de semejanza entre dos objetos. La base del comportamiento inteligente de un sistema radica en recordar situaciones similares existentes en el pasado (Gálvez Lio 2006).

Los SBC realizan el proceso de solución de problemas representando episodios de este

proceso. Un caso es un conjunto arbitrario de rasgos usados para describir un concepto particular. Los casos deben incluir tanto los éxitos como los fracasos para obtener esos objetivos. Los éxitos sugieren soluciones, los fracasos alertan de posibles errores.

La estructura de este tipo de sistemas se divide en cuatro pasos:

1. Recuperar los casos más parecidos donde un nuevo problema se aparea con casos similares guardados en la base de casos.
2. Reutilizar la solución propuesta en los casos para tratar de resolver el problema.
3. Revisar la solución propuesta.
4. Almacenar la nueva solución como parte de un nuevo caso.

Entre las ventajas que poseen los sistemas basados en casos se pueden encontrar que no requieren de un modelo explícito del dominio y el proceso de extracción se reduce a juntar casos históricos. La construcción de estos sistemas consiste en identificar atributos con los cuales describir los casos que permitan dar explicaciones. Usan técnicas de base de datos para manipular grandes volúmenes de información, pueden aprender adquiriendo nuevos conocimientos como casos haciendo su mantenimiento más fácil y pueden ir creciendo reflejando la experiencia acumulada (Arencibia Ramirez 2011).

A pesar del éxito de los sistemas basados en conocimiento, existen varios problemas entre los cuales se destacan: que el proceso de extracción de conocimiento es difícil, su construcción requiere de habilidades especiales, normalmente son lentos e incapaces de procesar grandes cantidades de información, son difíciles de mantener.

Esta técnica de inteligencia artificial según los estudios realizados por el autor, aporta mucho en el desarrollo de sistemas para la educación debido a que en estos resulta muy valioso para resolver problemas de aprendizaje en los estudiantes contar con la experiencia y utilización de soluciones a problemas similares resueltos con anterioridad. Su función básicamente es recuperar, adaptar y validar las soluciones encontradas almacenadas en la memoria. Usan técnicas de base de datos para manipular grandes volúmenes de información. Su relación con las condiciones del trabajo que se presenta está dada a partir de que el desempeño de cada estudiante puede tratarse como un caso.

1.4.2 Redes Neuronales

Las Redes Neuronales no son más que un modelo artificial y simplificado del cerebro humano. Es el ejemplo más perfecto del que se dispone para un sistema que es capaz de adquirir

conocimiento a través de la experiencia. Se comportan como un nuevo sistema para el tratamiento de la información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona (Martínez Labaut 2008).

Las redes neuronales son herramientas matemáticas para la modelación de problemas, que permiten obtener las relaciones funcionales subyacentes entre los datos involucrados en problemas de clasificación, reconocimiento de patrones, regresiones entre otros. Son consideradas excelentes aproximadoras de funciones esencialmente no lineales, siendo capaces de aprender las características relevantes de un conjunto de datos, para luego reproducirlas en entornos ruidosos o incompletos (Hammer 2003).

Una red neuronal puede definirse como un conjunto de unidades computacionales llamadas neuronas, interconectadas por medio de arcos pesados a manera de grafo dirigido. El objetivo de tal red, que puede verse como una caja negra, es calcular una salida “Y” a partir de una información “X” recibida con anterioridad. Usualmente las redes reciben la información proveniente del exterior mediante un conjunto de neuronas de entrada y cuentan con un conjunto distinto, llamado neuronas de salida, para ofrecer los resultados. El resto de las neuronas se organizan en capas ocultas (Arencibia Ramirez 2011) (Bonet Cruz 2008).

Se concibe el cálculo general de la red a partir de la información que es procesada por cada una de sus neuronas en forma independiente. Cada una de ellas puede recibir información de las restantes y calcular su propia salida a partir de dicha entrada y de su estado actual, transitando eventualmente hacia un nuevo estado. Por lo general, el flujo de cálculo de la red avanza progresivamente desde las neuronas de entrada hacia las neuronas de salida, en un proceso en el que cada una de las neuronas ocultas va activándose progresivamente según el esquema de conexión particular de cada red (Arencibia Ramirez 2011) (Bonet Cruz 2008).

Las Redes Neuronales Artificiales (RNA) son sistemas adaptativos que aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos. Las RNA crean su propia representación interna del problema, por tal motivo se dice que son auto organizadas. Se crean con la capacidad de responder adecuadamente cuando se les presentan situaciones a las que no habían sido expuestas anteriormente, es decir deben estar en condiciones de generalizar de casos anteriores a casos nuevos. Esta es una característica esencial debido que permite a la red responder correctamente no solo ante informaciones novedosas, sino también ante informaciones distorsionadas o incompletas (Martínez Labaut 2008).

El estudio de la técnica permite entender que con el uso de una RNA se intenta expresar la

solución de problemas complejos, mediante la evolución de unos sistemas de computación inspirados en el cerebro humano, y dotados por tanto de cierta inteligencia. Además el funcionamiento de la red puede verse como una caja negra, es decir a partir de una información de entrada recibida con anterioridad la red calcula una salida sin mostrar la forma de su funcionamiento interior.

1.4.3 Sistemas basados en Reglas

Los sistemas basados en reglas (SBR por sus siglas) son los más conocidos de los sistemas basados en el conocimiento (SBC). La forma de representación del conocimiento es a través de las reglas de producción. El proceso de solución de problemas en un SBR es crear una cadena de inferencias que constituye un camino entre la definición del problema y su solución. Esta cadena de inferencias puede construirse por dos vías o direcciones de búsqueda (Castillo 1996):

1. Comenzar con todos los datos conocidos y progresar hacia la conclusión (conocido en inglés como *data driven o forward chaining*).
2. Seleccionar una conclusión posible y tratar de probar su validez buscando evidencias que la soporten (conocido en inglés como *goal driven o backward chaining*).

Los sistemas basados en reglas trabajan mediante la aplicación de reglas, comparación de resultados y aplicación de las nuevas reglas basadas en una situación modificada. También pueden trabajar por inferencia lógica dirigida, bien empezando con una evidencia inicial en una determinada situación y dirigiéndose hacia la obtención de una solución, o bien con hipótesis sobre las posibles soluciones y volviendo hacia atrás para encontrar una evidencia existente (o una deducción de una evidencia existente) que apoye una hipótesis en particular (Martínez Labaut 2008).

Un sistema basado en reglas es muy eficaz cuando tiene que analizar gran cantidad de información, interpretándola y proporcionando una recomendación a partir de la misma, son buenos para predecir resultados futuros a partir del conocimiento que tienen. Es necesario dejar claro que un sistema basado en reglas, aunque es un tipo de sistema experto no sustituye al experto en sí. Los sistemas basados en reglas son una expresión de los sistemas basados en el conocimiento. Su función consiste en aportar soluciones a problemas o nuevo conocimiento (García Lorenzo 2012).

El diseño de un SBR requiere una enorme cantidad de conocimientos debido a que hay que tener en cuenta muchas especificaciones y restricciones. El SBR puede evaluar el nivel de

conocimiento y comprensión de un estudiante y ajustar el proceso de aprendizaje de acuerdo con sus necesidades.

Los SBR tienen ventajas y desventajas. Entre sus ventajas se tiene la fácil modificación pues el conocimiento es explícito y accesible, la consistencia en las respuestas, la gran accesibilidad ya que trabajan las 24 horas todos los días, ellos poseen capacidad para adquirir nuevo conocimiento y perfeccionar el que poseen, además explican las soluciones y tiene capacidad de expresar amplios conocimientos. Pero en ocasiones pueden traer problemas tales como el encadenamiento infinito, la adición de nuevo conocimiento que puede resultar contradictorio, así como la modificación de las reglas ya existentes (Martínez Labaut 2008).

Con el estudio de esta técnica se asume que la representación del conocimiento en estos sistemas se realiza a partir de reglas de producción, por lo cual aplicar la técnica resulta útil cuando se tiene de antemano experiencia. El método de solución utilizado es usualmente búsqueda primero en profundidad con dirección hacia adelante o hacia atrás y las fuentes de conocimiento son: expertos, publicaciones, ejemplos. Debido a que la solución de la presente investigación debe trabajar con datos sobre los cuales se aprende y de cuyo análisis se posibilitará la toma de decisiones, se descarta la utilización de los SBR en este aporte.

1.4.4 Sistemas basados en probabilidades

En los sistemas basados en probabilidades (SBP, por sus siglas) el conocimiento es representado a partir de probabilidades o frecuencias. Como método de solución se basa en el teorema de Bayes y otras técnicas de inferencia estadística teniendo como fuente de conocimiento los ejemplos.

Las redes Bayesianas son representadas por un gráfico acíclico, donde cada variable es representada por un nodo de la red y de ella se especifican dos tipos de información, la estructura de dependencias condicionales que son los arcos de la red y las distribuciones de probabilidad correspondientes. Una red Bayesiana puede calcular la distribución de probabilidad para cualquier subconjunto de variables de la red dado los valores o distribuciones de las variables restantes (Hammer 2003).

Una red bayesiana es un modelo gráfico probabilístico que representa un conjunto de variables y sus dependencias probabilísticas. Son grafos dirigidos, donde cada nodo representa un atributo. Estas redes pueden ser usadas para inferir un valor objetivo dado los valores observados de otras variables, y recíprocamente, inferir el valor probable de una variable, a partir de la evidencia de otras y/o del valor objetivo. Ello constituye el principal mérito de las

redes bayesianas (Arencibia Ramirez 2011).

La revisión realizada sobre esta técnica permitió comprobar que la forma de adquirir conocimiento de los SBP consiste en coleccionar muestras y realizar un procesamiento estadístico que produzca las probabilidades o frecuencias que forman la base de conocimientos. Por lo tanto se hace difícil construir las redes cuando existen carencias de conocimiento con ayuda de expertos humanos. Además no son viables para explicar el razonamiento, pues los métodos y modelos que utiliza están aún lejos de ofrecer explicaciones comprensibles y satisfactorias para los expertos que pudieran utilizar las redes desarrolladas actualmente. Por las razones expuestas con anterioridad no se sugiere su utilización para la investigación que se presenta.

Sobre los sistemas expertos estudiados anteriormente el autor asume para la presente investigación la utilización de la representación del conocimiento basado en casos, definiendo una estructura que permita representar la información. Cada caso estará formado por el nombre y apellidos del estudiante, el resultado de la evaluación inicial y la ruta de aprendizaje que siguió luego de haber obtenido una evaluación.



Figura 1. Representación de la estructura del conocimiento

Esta representación del razonamiento (Figura 1) ofrece la posibilidad de almacenar la información en la memoria del sistema durante la fase de aprendizaje conteniendo los problemas o casos ya resueltos. Pensando en una versión mejorada del propio sistema (SACOI-AA) podrá ser aplicada sobre esta base de casos una función de semejanza que determine los casos más parecidos de acuerdo con nuevos casos que se incorporen y las soluciones de estos casos recuperados se adaptan para actualizar la nueva solución que debe ofrecerse.

1.5 Estudio de las técnicas y herramientas que soportan la solución

Lenguaje de Programación

Existe una cierta cantidad de lenguajes, cada uno de ellos explota ciertas características que lo hacen más o menos útiles para los desarrolladores de software. Un lenguaje de programación aunque complicado en su aprendizaje permite en general informatizar una variedad de tareas que pudieran resultar complejas y tediosas si se realizan de forma manual. Es por eso que

cuando se va a elegir un lenguaje de programación es necesario saber si este puede aplicarse a una solución particular. De manera general la programación es un modo práctico para que los programadores puedan dar instrucciones a un equipo (Morejón González 2010).

1.5.1 PHP

El PHP (acrónimo de PHP: Hipertexto Pre-Processor), es un lenguaje script incrustado en HTML, de alto nivel y ejecutado en el servidor. La mayor parte de su sintaxis ha sido tomada de C y Java con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación de páginas dinámicas. Con PHP se puede realizar el procesamiento de información en formularios, foros de discusión, manipulación de cookies entre otras (Morejón González 2010).

Ventajas:

- Posee una comunidad muy grande de desarrolladores y existe abundante documentación, tutoriales, ejemplos de código, foros etc.
- PHP es fácil de aprender comparado con otros lenguajes de programación.
- El rendimiento de PHP es alto y verdaderamente eficiente.
- El precio para utilizar PHP es cero y se puede descargar desde www.php.net.
- Es Open Source lo cual quiere decir que se tiene acceso al código fuente.
- Está disponible para la mayoría de sistemas operativos existentes. Desde Unix, Linux, Microsoft Windows, MAC, entre otros.
- La versión 5 de PHP está diseñada para el soporte de características de programación como herencia, métodos y atributos públicos o privados, clases y métodos abstractos, constructores, interfaces y destructores orientada a objetos.
- Tiene soporte para conectarse a una gran variedad de bases de datos como: MySQL, PostgreSQL, Oracle.

Desventajas:

- El manejo de errores no es tan sofisticado como en otros lenguajes.
- PHP consume más recursos y por tanto es ligeramente lento llamando y ejecutando una función.

1.5.2 Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems con el cual se puede realizar cualquier tipo de programa. Desde la primera versión de java

existe la posibilidad de desarrollar pequeñas aplicaciones que luego pueden ser incrustadas en una página HTML para que sean descargadas y ejecutadas por el navegador web. Entre sus características se puede resaltar que: posee un mecanismo robusto de manejo eficiente de excepciones, tiene un modelo de objetos simple y elimina herramientas de bajo nivel, que suelen inducir muchos errores, como la manipulación directa de punteros o memoria, es multiplataforma y la utiliza el enlace dinámico en tiempo de ejecución (Gosling 2012).

Luego del análisis de los lenguajes de programación realizado anteriormente, se pudo contrastar las distintas características y facilidades de ellos, pero por las ventajas que brinda el lenguaje PHP principalmente en su versión 5.3.8 es que se decide su uso para la implementación del sistema que en esta investigación se propone.

Gestor de Bases de Datos

Un sistema gestor de base de datos (SGBD) se puede definir como el conjunto de programas que administran y gestionan la información contenida en una base de datos. De manera general los sistemas gestores de bases de datos posibilitan realizar las siguientes acciones:

- Definición de la información.
- Mantenimiento de la integridad de los datos dentro de la base de datos.
- Control de la seguridad y privacidad de los datos.
- Manipulación de los datos.

1.5.3 PostgreSQL

Es un sistema de gestión de bases de datos relacional orientada a objetos y libre. El desarrollo de PostgreSQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales los cuales trabajan en su desarrollo (Morejón González 2010) (Herrera Chica 2012).

Ventajas:

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP) y cadenas de bits. También permite la creación de tipos propios.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes.
- Incluye herencia entre tablas, por lo que se incluye entre los gestores de objetos-relacionales.

- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Posee gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, siendo capaz de soportar gran cantidad de peticiones simultáneas de manera correcta.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel.
- Es un producto Open Source, sin costos de licencia.

Desventajas:

- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- La velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes, característica que resulta loable.

1.5.4 MySQL

Es un SGBD relacional que almacena la información en tablas separadas en vez de ponerlas todas en un gran almacén; añadiéndole velocidad y flexibilidad en su funcionamiento. Se encuentra liberado bajo la licencia GPL (General Public License conocido en inglés). MySQL es catalogado como el gestor más rápido de los conocidos. Puede trabajar en modo cliente/servidor o en sistemas embebidos (Herrera Chica 2012).

Las principales características de MySQL son:

- Trabaja en varias plataformas.
- Las funciones SQL están implementadas utilizando una librería de clases optimizada tan rápida como sea posible. Usualmente no hay asignación de memoria.
- Provee motores de almacenamientos transaccionales y no transaccionales.

Luego de realizar este estudio se decide tomar a PostgreSQL 9.1 como el sistema gestor de base de datos ya que por las características y ventajas que brinda es más amplia su utilización. Además es el SGBD del cual se tiene mayor información pues existen grandes comunidades de desarrolladores las cuales tienen un gran conocimiento de sus características. Finalmente es de destacar su adaptabilidad a las prestaciones que brinda el sistema de la plataforma donde se vaya a utilizar.

Framework de Desarrollo

El concepto de framework no solo se emplea en aplicaciones web sino que también está en muchos ámbitos del desarrollo de sistemas de software. Se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las piezas necesarias para construir una aplicación completa. Con el framework se hace referencia a una estructura de software compuesta de varios componentes personalizables e intercambiables para desarrollar una aplicación. Son soluciones que implementan patrones y ayudan a desarrollar funciones dentro de las aplicaciones a un nivel superior (Morejón González 2010) (Duarte Correa 2010).

Los objetivos principales que persigue un framework son:

- Acelerar el proceso de desarrollo.
- Reutilizar código ya existente.
- Promover buenas prácticas de desarrollo como el uso de patrones.

1.5.5 Zend Framework

Zend Framework se trata de un framework para desarrollo de aplicaciones y servicios web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además es Open Source y trabaja con PHP 5. Sus principales características son:

- Trabaja con el patrón Modelo Vista Controlador (MVC por sus siglas en inglés).
- Cuenta con módulos para manejar archivos .pdf.
- El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.
- Para el acceso a base de datos, balancea el ORM con eficiencia y simplicidad.
- Robustas clases para autenticación y filtrado de entrada.

1.5.6 Symfony

Symfony es un framework para crear aplicaciones PHP y tiene una forma sencilla de aumentar la productividad y la calidad de las aplicaciones web desarrolladas. Symfony ha sido probado con éxito en algunos de los sitios web más grandes del mundo. Utiliza patrones de diseño que se han definido para la web (Morejón González 2010) (Duarte Correa 2010).

También es fácil de instalar y configurar en sistemas Windows, Mac y Linux. Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server). Es

compatible con PHP 5, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP. Flexible hasta cualquier límite y extensible mediante un completo mecanismo de Plugins y traducido a más de 40 idiomas (Morejón González 2010).

Ventajas:

- Independiente del sistema gestor de bases de datos.
- Sencillo de usar y flexible como para adaptarse a los casos más complejos.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Fácil de extender permitiendo su integración con librerías desarrolladas por terceros.

Desventajas:

- Fuerte curva de aprendizaje, aunque la documentación disponible facilita esta tarea.
- Todavía hay muchas aplicaciones y desarrolladores que programan en PHP4. Esto puede plantear un problema ya que para extraer toda la potencia del framework, el programador debe tener un buen conocimiento del paradigma de programación orientado a objetos.
- Aumenta el peso de la aplicación debido a que la implementación del patrón MVC introduce varias capas de objetos que debe procesar el intérprete.

A partir del estudio realizado se decide tomar a *Symfony2* como el framework a utilizar en el desarrollo de la aplicación ya que es sencillo de usar en la mayoría de los casos, además es flexible como para adaptarse a los casos más complejos, es adaptable a las políticas y arquitecturas propias de cada empresa y ha sido probado con éxito en algunos de los sitios web más grandes del mundo. Utiliza patrones de diseño que se han definido para la web, es decir que complementa el trabajo a realizar con estas ventajas brindadas por el framework.

Metodología de desarrollo

Con el acelerado avance de las TIC el desarrollo de software se vuelve cada vez más complejo y los clientes demandan soluciones cada día más potentes. El uso de las metodologías de desarrollo de software se justifica en la necesidad de que los desarrolladores posean un único lenguaje de entendimiento, el cual influye en la calidad del trabajo y a su vez en la satisfacción del cliente en cuanto al resultado. (Brito Acuña 2009)

Las metodologías de desarrollo de software se han clasificado en dos grandes grupos: las ágiles y las tradicionales o pesadas. Las metodologías ágiles enfatizan la comunicación con el

cliente mientras que suelen ser criticadas por la falta de documentación técnica, generalmente propone que los involucrados participen en el proceso de desarrollo para ir mejorando las funcionalidades del sistema en producción. Las tradicionales tienen mayor énfasis en la planificación y control del proyecto, en la especificación precisa de requisitos y modelado, son generalmente aplicadas a grandes proyectos e indican paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben detener.

Entre las metodologías ágiles se destacan XP (*eXtreme Programming*) y Crystal. Así mismo se puede mencionar RUP (*Rational Unified Process*) y MSF (*Microsoft Solution Framework*) como metodologías pesadas.

1.5.7 Extreme Programing

La Programación Extrema (XP por sus siglas en inglés) se ha convertido en la metodología ágil más popular. Su creador fue Kent Beck, impulsor del manifiesto ágil. XP consiste en una programación rápida o extrema, cuyo rasgo característico es tener como parte del equipo de desarrollo al cliente que es el usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas. Esta programación considera que con un poco de planificación, codificación y unas pruebas, se puede decidir si se está siguiendo el camino correcto o no, evitando esfuerzos innecesarios (Brito Acuña 2009) (Herrera Chica 2012). Su basamento teórico se justifica en los siguientes elementos:

Pruebas Unitarias o Caja Blanca: Pruebas realizadas a los principales procesos.

Refabricación: Reutilización de código a partir de crear patrones o modelos estándares que traen consigo una mayor flexibilidad al cambio.

Programación en pares: Dos desarrolladores participan en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.

El ciclo de desarrollo consiste en los siguientes pasos:

- El cliente define el valor de negocio a implementar.

- El programador estima el esfuerzo necesario para su implementación.
- El cliente selecciona qué construir, de acuerdo a sus prioridades y las restricciones de tiempo.
- El programador construye ese valor de negocio.
- Vuelve al paso 1.

El ciclo de vida ideal de XP consta de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Cierre del Proyecto (Brito Acuña 2009).

1.5.8 Rational Unified Process

Rational Unified Process (RUP conocido en inglés), es una plataforma de procesos de desarrollo de software flexible que ayuda brindando guías consistentes y personalizadas de procesos para todo el equipo de proyecto. RUP describe cómo utilizar de forma efectiva reglas de negocio y procedimientos comerciales probados en el desarrollo de software. Captura varias de las mejores prácticas en el desarrollo moderno de software en una forma que es aplicable para un amplio rango de proyectos y organizaciones. Es una guía de cómo utilizar de manera efectiva UML (conocido en inglés como *Unified Modeling Language*). Provee a cada miembro del equipo fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo. Crea y mantiene modelos, en lugar de enfocarse en la producción de una gran cantidad de papeles de documentación (Brito Acuña 2009).

Cubre el ciclo de vida de desarrollo de software. RUP presenta 3 características esenciales:

Dirigidos por casos de uso: Los casos de uso reflejan lo que los futuros usuarios necesitan y desean, estos representan los requisitos funcionales. Los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

Centrado en la Arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de trabajo y los usuarios deben de estar de acuerdo, por lo que describe los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

Iterativo e incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo y los incrementos, al crecimiento del producto (Herrera Chica 2012).

Una de las mejores prácticas centrales de RUP es la noción de desarrollar iterativamente. RUP organiza los proyectos en términos de disciplinas y fases, consistiendo cada una en una o más iteraciones. Con esta aproximación iterativa, el énfasis de cada flujo de trabajo variará a través del ciclo de vida. La metodología RUP divide en 4 fases el desarrollo del software:

- Inicio: Determinar la visión del proyecto.
- Elaboración: Determina la arquitectura óptima.
- Construcción: Obtiene la capacidad operacional inicial.
- Transmisión: Obtiene la liberación del proyecto.

Cada una de estas fases se desarrolla mediante el ciclo de iteraciones, las cuales tienen sus objetivos en función de la evaluación de las iteraciones precedentes. El ciclo de vida que se desarrolla por cada iteración es llevada bajo dos disciplinas:

1. Disciplina de Desarrollo: Modelación del Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas.
2. Disciplina de Soporte: Despliegue, Administración de configuración y cambios, Administración de Proyecto, Ambiente.

Luego del estudio de las metodologías más utilizadas mundialmente, se considera que la metodología RUP garantiza la elaboración de todas las fases de un producto software orientado a objeto. Permite realizar la documentación del software durante todo el proceso de desarrollo, pero posee como inconveniente que es para desarrollar proyectos de gran envergadura. Por su parte XP es una metodología ágil que sirve para proyectos de corto plazo, su documentación es ligera y su curva de aprendizaje es suave, por lo que la hace eficiente y flexible en el caso de la presente investigación. Por otra parte la programación extrema ha surgido desde la experiencia, como una forma de resolver los problemas encontrados en los procesos de desarrollo software en los que se han visto involucrados sus autores. XP es mucho más fácil de implementar y de aprender. Mientras que en RUP la entrega final debe de ser algo mucho más definido en XP se realizan entregas continuas y discretas que permiten evaluar el sistema por lo que se puede concluir que XP presenta muchos elementos que pueden ser útiles para el desarrollo del sistema.

Herramientas Case

La realización de un nuevo software requiere que las tareas sean organizadas y completadas en forma correcta y eficiente. Las herramientas CASE se utilizan para ayudar a automatizar los procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el

ciclo de desarrollo de un producto de software. Estas se pueden definir como un conjunto de programas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software (Duarte Correa 2010) (Herrera Chica 2012).

1.5.9 Suite Visual Paradigm

La Suite de productos que propone el Visual Paradigm se utiliza para desarrollar software de manera eficiente, rápida y de forma colaborativa. Permite realizar diagramas de procesos de negocios, modelado UML, modelos de casos de usos, modelos de actividad, de interacción, de bases de datos, de entidad-relación. Además posee integración para varios IDE y en ella también se puede generar documentación bajo un modelo colaborativo. Visual Paradigm soporta todas las necesidades de diseño y modelado a lo largo del ciclo de vida de desarrollo de software, es una herramienta que ayuda a construir aplicaciones de calidad, de manera más rápida, óptima y barata (Herrera Chica 2012).

Considerando todas las facilidades brindadas por la Suite Visual Paradigm, que es un producto de calidad, muy fácil de instalar y actualizar, que está disponible en varias ediciones entre las que se encuentran Enterprise, Professional, Community, Standard, Modeler y Personal, ofreciéndoles a los usuarios la posibilidad de escoger según sus necesidades; y que además, es una herramienta de software libre de la cual se cuenta con una licencia en la universidad, se puede concluir que constituye la mejor opción para realizar el modelado de diagramas de este trabajo según la metodología propuesta.

1.6 Conclusiones parciales

A partir de lo investigado durante el estudio realizado y presentado en el capítulo, se puede concluir lo siguiente:

- El CC como resultado de la implementación continua de las tecnologías hace posible en el campo de la educación y específicamente en la enseñanza universitaria que se fortalezcan el auto aprendizaje, la auto gestión del conocimiento y de manera general se logre una mayor independencia de los estudiantes durante el proceso de enseñanza-aprendizaje.
- Se determinó que el diseño de una secuencia de tareas docentes, interrelacionadas y/o complementadas, que tiene el fin de cumplir con uno o varios objetivos instructivos se le denominará “ruta de aprendizaje”.
- Se determinó el uso de las analíticas de aprendizaje para guiar el proceso de análisis de

la información que se obtiene de los estudiantes durante el proceso docente.

- Se seleccionó la representación del conocimiento basados en casos y se definió su estructura de representación.
- Se decidió utilizar como lenguaje de programación PHP, como *framework* de desarrollo *Symfony2*, como metodología de desarrollo XP y como herramienta para el desarrollo del modelado Visual Paradigm. Todas estas herramientas, lenguaje y metodología brindan las prestaciones necesarias para llevar a cabo el desarrollo de un sistema con las características de SACOI-AA con una estructura modular y sencilla.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

El presente capítulo tiene como objetivo definir los detalles técnicos de SACOI-AA. Se presentan y describen los principales artefactos generados al utilizar la metodología XP, la cual fue seleccionada para llevar a cabo el desarrollo de la aplicación. El capítulo se enfoca en la descripción de las características generales del sistema, elemento que se concreta en la captura y especificación de los requisitos.

2.1 Propuesta de sistema

Descripción general de la propuesta de sistema

El sistema será capaz de analizar el cumplimiento de objetivos instructivos para los estudiantes matriculados en un determinado curso utilizando analíticas de aprendizaje. Los usuarios típicos del sistema son estudiante, profesor, profesor principal y administrador. Para utilizar el sistema el usuario deberá autenticarse en la aplicación, esta verificará sus privilegios, permitiéndole acceder solo a las áreas, funcionalidades y acciones que le están permitidas según su rol. A los usuarios con rol administrador les permitirá realizar una gestión general del perfil de los estudiantes, de los profesores y de la asignación de estudiantes y profesores a los grupos de clases. A los profesores principales les permitirá introducir la documentación necesaria para cada tema de la asignatura, así como la creación de la ruta de aprendizaje de una asignatura, crear evaluaciones que respondan a la comprobación de los objetivos de la asignatura. El sistema permitirá emitir un reporte del nivel de completamiento de los objetivos instructivos de la asignatura de acuerdo a los contenidos vencidos y a las habilidades logradas.

Fase de Exploración

2.2 Captura de los requisitos funcionales de software

Especificación de los Requisitos del Software

La intención general de la captura de requisitos es obtener una descripción correcta de lo que el sistema debe hacer y delimitar su alcance. Los requisitos tienen un papel importante durante el ciclo de vida de una aplicación porque se enfocan en la descripción del propósito del sistema. Funcionan como un contrato entre el cliente y los desarrolladores. Los requisitos en su mayoría se identifican durante la fase de inicio y se clasifican en requisitos funcionales y no funcionales.

Requisitos Funcionales

Los requisitos funcionales especifican las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Son capacidades o condiciones que el sistema debe cumplir.

El sistema para el análisis de cumplimiento de objetivos instructivos utilizando técnicas de Analíticas de Aprendizaje cuenta con los siguientes requisitos funcionales, los cuales se encuentran agrupados según sus funcionalidades:

Requisitos Funcionales del módulo de Administración

RF1-Establecer los permisos correspondientes para la modificación y aprobación de los cambios a nivel de estudiante y profesor.

RF2-Gestionar el perfil de los estudiantes de la asignatura.

RF2.1-Adicionar los datos del perfil de un nuevo estudiante.

RF2.2-Modificar los datos del perfil de un estudiante existente.

RF2.3-Eliminar los datos del perfil de un estudiante existente.

RF2.4-Mostrar los datos del perfil de todos los estudiantes existentes.

RF3-Gestionar el perfil de los profesores de la asignatura.

RF3.1-Adicionar los datos del perfil de un nuevo profesor.

RF3.2--Modificar los datos del perfil de un profesor existente.

RF3.3-Eliminar los datos del perfil de un profesor existente.

RF3.4-Mostrar los datos del perfil de todos los profesores existentes.

RF4-Gestionar las asignaturas.

RF4.1-Adicionar los datos de una nueva asignatura.

RF4.2-Modificar los datos de una asignatura existente en el sistema.

RF4.3-Eliminar los datos de una asignatura existente en el sistema.

RF4.4-Mostrar todos los datos de las asignaturas existentes en el sistema.

RF5-Gestionar las facultades.

RF5.1-Adicionar nuevas facultades.

RF5.2-Modificar los datos de una facultad existente en el sistema.

RF5.3-Eliminar los datos de una facultad existente en el sistema.

RF5.4-Mostrar todos los datos de una facultad existente en el sistema.

RF6-Gestionar los grupos de una facultad.

RF6.1-Adicionar los datos de los nuevos grupos de la facultad.

RF6.2-Modificar los datos de un grupo existente en el sistema.

RF6.3-Eliminar los datos de un grupo existente en el sistema.

RF6.4-Mostrar todos los datos de un grupo existente en el sistema.

RF7-Enlazar estudiantes a los grupos asignados a un profesor de la asignatura.

RF8-Gestionar los objetivos y habilidades de las asignaturas.

RF8.1-Adicionar los objetivos y habilidades de la nueva asignatura.

RF8.2-Modificar los objetivos y habilidades de la asignatura existente en el sistema.

RF8.3-Eliminar los objetivos y habilidades de la asignatura existente en el sistema.

RF8.4-Mostrar todos los objetivos y habilidades de la asignatura existente en el sistema.

Requisitos Funcionales del módulo de Diseño Instruccional

RF9-Gestionar el diseño Instruccional de una asignatura.

RF9.1-Adicionar la ruta de aprendizaje de una asignatura.

RF9.2-Modificar la ruta de aprendizaje de una asignatura.

RF9.3-Eliminar la ruta de aprendizaje de una asignatura.

RF9.4- Mostrar todos los datos de la ruta de aprendizaje de una asignatura.

RF10-Crear evaluaciones de acuerdo a las habilidades y que verifiquen el grado de cumplimiento de cada objetivo instructivo de la asignatura.

RF11-Graficar la ruta a seguir según el nivel de asimilación de cada estudiante por cada contenido de la asignatura de acuerdo al tiempo entre una actividad docente y la siguiente.

RF12-Emitir un reporte del nivel de completamiento de los objetivos instructivos de la asignatura de acuerdo a los contenidos vencidos y a las habilidades logradas (clases prácticas resueltas, evaluaciones realizadas).

2.3 Descripción de los requisitos no funcionales de software

Los requisitos no funcionales (RNF) de la solución son características y prestaciones que se han de tener en cuenta para la implementación del futuro sistema. Son condiciones que tendrán las estaciones de trabajo y servidores para que el software final muestre un cumplimiento satisfactorio de las funcionalidades del sistema. En conjunto con los requisitos funcionales se logra alcanzar el desarrollo y funcionamiento de la aplicación.

En el caso de SACOI-AA se tiene que los RNF organizados por características técnicas son los siguientes:

Requisitos de Seguridad

La información manejada por el sistema podrá ser accedida por las personas que tienen los permisos para ello, por lo que estará protegida de acceso no autorizado y divulgación. Los usuarios accederán solo a la información correspondiente a su rol.

Requisitos de Integridad

Los datos contenidos en la Base de Datos no deberán ser redundantes o falsos. La información generada por el sistema será objeto de un alto nivel de protección contra la corrupción.

Requisitos de Disponibilidad

A los usuarios autorizados se les deberá garantizar el acceso a la información en cualquier momento que esta sea solicitada.

Usabilidad

La aplicación que se utilizará debe ser lo más interactiva posible, brindará una interfaz simple y amigable para que el usuario no tenga dificultad al utilizarla.

Rendimiento

El sistema debe ser lo más eficiente posible para poder lograr un tiempo de respuesta de 0.1 a 0.7 segundos ante la concurrencia de peticiones.

Portabilidad

La aplicación podrá ser usada en sistemas operativos Windows o Linux.

Confiabilidad

Se garantizará la llegada de los datos de forma íntegra y segura al destino. Los datos serán almacenados de forma segura en una Base de Datos.

Requisitos de Software

- El servidor central de procesamiento deberá tener instalado como Sistema Operativo Windows o Linux, deberá contar con el Framework *Symfony2*, así como con un servidor de aplicaciones web como el servidor WAMP 2.0 (Windows) o LAMP (Linux).
- Las PC clientes deberán tener instalado como Sistema Operativo Windows o Linux. Además deberá contar con navegadores Web Internet Explorer o Mozilla Firefox.

- El servidor central de bases de datos deberá tener instalado Sistema Operativo Windows o Linux y Gestor de Base de Datos Postgre SQL 8.3.

Requisitos de Hardware

- Todas las computadoras deberán tener una tarjeta de red con una velocidad de transmisión de 100 Mbps.
- El Servidor central de procesamiento requiere de memoria RAM de 1 GB DDR2 o mayor, un procesador de 1.90 GHZ, 80 GB de capacidad de almacenamiento mínimo o mayor.
- Las PC clientes requieren de memoria RAM 128 mega byte DDR2, un micro procesador de 1.90 GHZ.
- El servidor central de bases de datos requiere una memoria RAM de 1 GB DDR2, un micro procesador de 1.90 GHZ y 80 GB de capacidad de almacenamiento mínimo.

2.4 Descripción de las historias de los usuarios

Entre los artefactos generados por la metodología XP se pueden encontrar las historias de usuarios (HU). Las HU son creadas a partir de las funcionalidades definidas. En las siguientes tablas se mostrarán algunas de las HU de los diferentes Requisitos Funcionales (RF), que presentan prioridad alta por constituir principales funcionalidades del sistema. Las HU mostradas a continuación se corresponden con las funcionalidades “9”, “10”, “11”, “12”. Las HU correspondientes se muestran en una plantilla que ha sido modificada por el autor en función de dejar la información estrictamente necesaria. Los prototipos de interfaz de usuario correspondiente a cada HU pueden encontrarse en los anexos.

Se realiza la descripción de las historias de los usuarios con sus tareas ingenieriles y su responsable asignado para el cumplimiento de las mismas (Ver Tabla 1, Tabla 2; Tabla 3; Tabla 4; Tabla 5; Tabla 6; Tabla 7; Tabla 8).

Tabla 1: Descripción de la historia de usuario “Adicionar la ruta de aprendizaje de una asignatura”.

Historia de Usuario	
Código: HU27	Nombre Historia de Usuario: Adicionar la ruta de aprendizaje de una asignatura.

Modificación de Historia de Usuario Número: Ninguna	
Referencia: RF9.1	
Programador: Yasmany Febles Espinosa	Iteración Asignada: Primera
Prioridad: Alta	Puntos Estimados: 1 Semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1 Semana
Descripción: Permite introducir en el sistema la documentación necesaria para luego crear los nodos que conforman la ruta de aprendizaje y luego enlazarlos en el orden lógico por el que deben ir transitando los estudiantes.	
Observaciones: 1. Debe estar disponible la documentación necesaria.	
Prototipo de interface: Figura 19	

Tabla 2: Descripción de la historia de usuario “*Modificar la ruta de aprendizaje de una asignatura*”.

Historia de Usuario	
Código: HU28	Nombre Historia de Usuario: Modificar la ruta de aprendizaje de una asignatura.
Modificación de Historia de Usuario Número: Ninguna	
Referencia: RF9.2	
Programador: Yasmany Febles Espinosa	Iteración Asignada: Primera
Prioridad: Alta	Puntos Estimados: 1 Semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1 Semana
Descripción Permite modificar en el sistema la documentación necesaria para luego actualizar los nodos que conforman la ruta de aprendizaje y luego enlazarlos en el orden lógico por el que deben ir transitando los estudiantes.	
Observaciones: 1. Debe estar disponible la documentación necesaria.	
Prototipo de interface: Figura 19	

Tabla 3: Descripción de la historia de usuario “*Eliminar la ruta de aprendizaje de una asignatura*”.

Historia de Usuario	
Código: HU29	Nombre Historia de Usuario: Eliminar la ruta de aprendizaje de una asignatura.
Modificación de Historia de Usuario Número: Ninguna	
Referencia: RF9.3	

Programador: Yasmany Febles Espinosa	Iteración Asignada: Primera
Prioridad: Alta	Puntos Estimados: 1 Semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1 Semana
Descripción: Permite eliminar del sistema la documentación necesaria para luego actualizar los nodos que conforman la ruta de aprendizaje y luego enlazarlos en el orden lógico por el que deben ir transitando los estudiantes.	
Observaciones:	
Prototipo de interface: Figura 19	

Tabla 4: Descripción de la historia de usuario “Mostrar la ruta de aprendizaje de una asignatura.”

Historia de Usuario	
Código: HU30	Nombre Historia de Usuario: Mostrar la ruta de aprendizaje de una asignatura.
Modificación de Historia de Usuario Número: Ninguna	
Referencia: RF9.4	
Programador: Yasmany Febles Espinosa	Iteración Asignada: Primera
Prioridad: Alta	Puntos Estimados: 1 Semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1 Semana
Descripción: Permite mostrar en el sistema la documentación necesaria en los nodos que conforman la ruta de aprendizaje que deben ser transitados por los estudiantes.	
Observaciones:	
Prototipo de interface: Figura 19	

Tabla 5: Descripción de la historia de usuario “Crear evaluaciones que permitan comprobar el cumplimiento de los objetivos instructivos”.

Historia de Usuario	
Código: HU31	Nombre Historia de Usuario: Crear evaluaciones que comprueben las habilidades y que verifiquen el grado de cumplimiento de cada objetivo instructivo de la asignatura.
Modificación de Historia de Usuario Número: Ninguna	
Referencia: RF10	
Programador: Yasmany Febles Espinosa	Iteración Asignada: Primera
Prioridad: Alta	Puntos Estimados: 1 Semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1 Semana

Descripción: Permite al usuario con rol profesor principal cuando monte una asignatura, tener la posibilidad de montar una actividad que sirva de evaluación inicial para en correspondencia con el resultado obtenido por el estudiante sugerirle la Ruta de aprendizaje con la cual debería obtener los conocimientos que lo guíen en el cumplimiento de los objetivos instructivos de la asignatura.

Observaciones:

1. No es obligatoria la evaluación inicial, el profesor puede iniciar la ruta con el material que considere necesario para introducir un tema.
2. El profesor principal debe tener la posibilidad de crear rutas de aprendizajes para todo tipo de estudiantes, los cuales pueden obtener alta puntuación, puntuaciones medias y baja puntuación.

Prototipo de interface: Figura 20

Tabla 6: Descripción de la historia de usuario “Graficar la ruta de aprendizaje de cada estudiante”.

Historia de Usuario	
Código: HU32	Nombre Historia de Usuario: Graficar la ruta a seguir según el nivel de asimilación de cada estudiante por cada contenido de la asignatura de acuerdo al tiempo entre una actividad docente y la siguiente.
Modificación de Historia de Usuario Número: Ninguna	
Referencia: RF11	
Programador: Yasmany Febles Espinosa	Iteración Asignada: Primera
Prioridad: Alta	Puntos Estimados: 1 Semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1 Semana
Descripción: Permite mostrar la Ruta de aprendizaje correspondiente al resultado alcanzado en la evaluación inicial por el estudiante, mediante la cual transitara por la asignatura, con la correspondiente documentación de los contenidos que deberá utilizar.	
Observaciones:	
<ol style="list-style-type: none"> 1. El sistema solo mostrará el nodo con la documentación necesaria donde se encuentra el estudiante en ese momento. 2. El profesor tendrá la posibilidad de observar el recorrido de sus estudiantes por las rutas transitadas. 	
Prototipo de interface: Figura 21	

Tabla 7: Descripción de la historia de usuario “Emitir reporte del nivel de completamiento de los objetivos vencidos”.

Historia de Usuario	
Código: HU33	Nombre Historia de Usuario: Emitir un reporte del nivel de completamiento de los objetivos instructivos de la asignatura de acuerdo a los contenidos vencidos y a las habilidades logradas.
Modificación de Historia de Usuario Número: Ninguna	

Referencia: RF12	
Programador: Yasmany Febles Espinosa	Iteración Asignada: Primera
Prioridad: Alta	Puntos Estimados: 1 Semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1 Semana
Descripción: Permite mostrarle al usuario registrado con rol profesor el avance del cumplimiento de los objetivos por parte de sus estudiantes.	
Observaciones:	
Prototipo de interface: Figura 23	

Realización del diseño

Los estilos, patrones arquitectónicos y patrones de diseño, representan, de lo general a lo particular, los niveles de abstracción y composición que tendrá la arquitectura del sistema. En este sentido, a continuación se definen los que soportan la implementación de la propuesta de solución que se ofrece en la presente investigación.

2.5 Estilo Arquitectónico

Los estilos arquitectónicos están encargados de describir la estructura general de un sistema, además de definir los componentes del mismo, su relación e interactividad. Se utilizan además para sintetizar estructuras de soluciones, o sea, tener un lenguaje que describa la estructura de la solución que luego será refinada a través del diseño. Toda estructura de una aplicación corresponde a un estilo bien conocido (Canales Mora 2009). Los estilos arquitectónicos definen los patrones arquitectónicos posibles de las aplicaciones. La clave del trabajo arquitectónico se encuentra en la elección de un estilo adecuado.

Estilos de Llamada y Retorno

Esta familia de estilos enfatiza la modificabilidad y la escalabilidad. Son los estilos más generalizados en sistemas en gran escala. Miembros de la familia son las arquitecturas de programa principal y subrutina, los sistemas basados en llamadas a procedimientos remotos, los sistemas orientados a objetos, los sistemas jerárquicos en capas y los sistemas con arquitectura Modelo-Vista-Controlador.

Estilo Modelo Vista Controlador (MVC)

Es el estilo de arquitectura de software que se debe seguir cuando se desarrolla una aplicación web, lo implementan una serie de *framework* de desarrollo web, en PHP u otros lenguajes como es el caso de *Symfony2*. Es interesante porque separa en varios grupos las complejidades de las distintas partes que componen una aplicación web. Separa los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes diferentes: Modelo, Vista y Controlador.

2.6 Patrón Arquitectónico

Es el nivel en el cual la arquitectura de software define la estructura básica de un sistema, pudiendo estar relacionado con otros patrones. Constituyen plantillas para la construcción de arquitecturas de software concretas, que provee un conjunto de subsistemas aportando las normas para su organización (Welicki 2009). La selección de un patrón de arquitectura o la combinación de varios es el paso más importante cuando se diseña la arquitectura de un sistema software.

Los patrones arquitectónicos son los relacionados con la interacción de objetos dentro o entre los niveles arquitectónicos. Se aplican durante la fase de diseño inicial para resolver problemas arquitectónicos, adaptabilidad a requerimientos cambiantes, rendimiento, modularidad y acoplamiento.

Patrón arquitectónico Modelo Vista Controlador

Este patrón representado en la Figura 2 separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

Modelo: Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador). Representa la información con la que trabaja la aplicación, es decir, su lógica del negocio.

Vista: Maneja la visualización de la información que se le mostrará a los usuarios. Presenta el modelo en un formato adecuado, como en una página web que le permite al usuario interactuar con ella.

Controlador: Responde a eventos, usualmente acciones del usuario. Interpreta las acciones del mouse y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado. (Ver Figura 2)

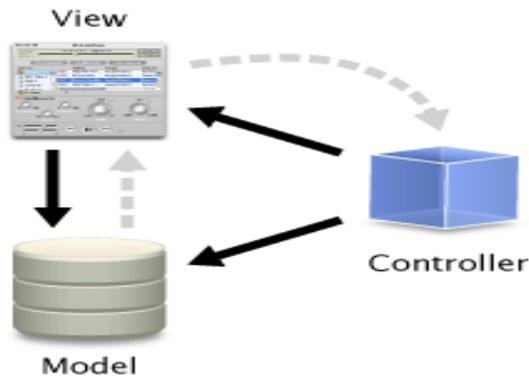


Figura 2: Representación del patrón arquitectónico MVC.

Symfony2 basa su funcionamiento interno en la famosa arquitectura Modelo - Vista -Controlador (MVC), utilizada por la mayoría de los *framework* web. No obstante, según su creador Fabien Potencier: *Symfony2* no es un framework MVC ya que sólo proporciona herramientas para las partes del Controlador y de las Vistas. La parte del Modelo es responsabilidad del programador, aunque provee librerías para integrar fácilmente los ORM más conocidos, como *Doctrine2* y *Propel* (Eguiluz 2011).

Observa en la siguiente figura (Ver Figura 3) simplificada el funcionamiento interno de *Symfony2*, el mismo toma lo mejor de la arquitectura MVC y la implementa de forma tal que el desarrollo de aplicaciones sea rápido y sencillo.

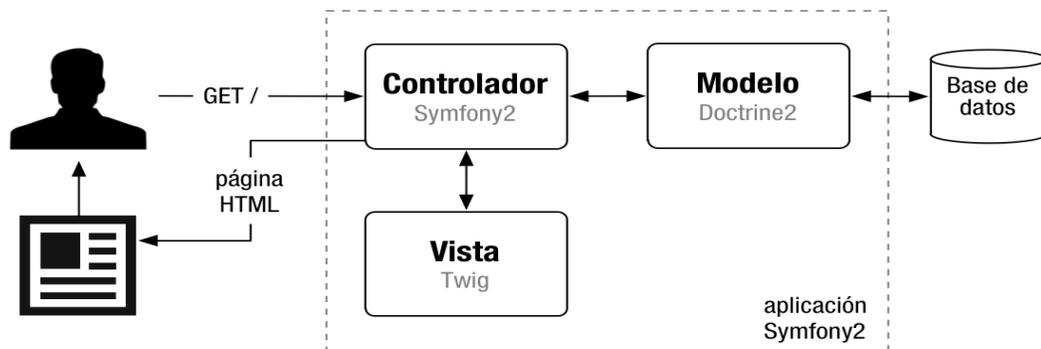


Figura 3: Representación del patrón MVC que implementa *Symfony2*.

Cuando el usuario solicita ver la portada del sitio, internamente sucede lo siguiente:

1. El sistema de enrutamiento determina qué **Controlador** está asociado con la página de la portada.

2. *Symfony2* ejecuta el **Controlador** asociado a la portada. Un controlador no es más que una clase PHP en la que se puede ejecutar cualquier código.
3. El **Controlador** solicita al **Modelo** los datos que se le están solicitando. El modelo no es más que una clase PHP especializada en obtener información, normalmente de una base de datos.
4. Con los datos devueltos por el **Modelo**, el **Controlador** solicita a la **Vista** que cree una página mediante una plantilla y que inserte los datos del **Modelo**.
5. El **Controlador** entrega al servidor la página creada por la **Vista**.

A pesar de que se puede llegar a hacer cosas muy complejas con *Symfony2*, el funcionamiento interno siempre es el mismo:

- 1) El **Controlador** manda y ordena.
- 2) El **Modelo** busca la información que se le solicita.
- 3) La **Vista** crea páginas con plantillas y datos.

2.7 Patrones de Diseño

Los patrones de diseño se consideran el tercer nivel de abstracción de la arquitectura de software, cuya finalidad es la de precisar en detalles los subsistemas y componentes de la aplicación.

Describen la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular. Los mismos se aplican durante la fase de diseño detallado para solucionar problemas de claridad del diseño, multiplicación de clases y adaptabilidad a requerimientos cambiantes (Tedeschi 2011).

2.7.1 Patrones GRASP

GRASP (*General Responsibility Assignment Software Patterns*) en sus siglas llevadas al español corresponde a los Patrones Generales de Software de Asignación de Responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Welicki 2009). A continuación se describen los patrones básicos de asignación de responsabilidades utilizados en el desarrollo del sistema:

- **Experto:** Es uno de los patrones que más se utiliza cuando se trabaja con *Symfony2*, con la inclusión de la librería *Doctrine2* para mapear la Base de Datos. *Symfony2* utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades. Las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan.
- **Creador:** En la clase *Actions* se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase *Actions* es “creador” de dichas entidades.
- **Alta Cohesión:** *Symfony2* permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase *Actions*, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades. *Symfony2* agrupa las clases por funcionalidades que son fácilmente reutilizables, bien por su uso directo o por herencia.
- **Bajo Acoplamiento:** Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja. A cada clase *Symfony2* le asigna una responsabilidad, de forma tal que mantiene pocas dependencias entre las mismas.
- **Controlador:** Todas las peticiones Web son manipuladas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en las clases *DefaultController.php* del Módulo de Administración, *DefaultController.php* del Módulo de Diseño Instruccional. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

2.7.2 Patrones GOF

La familia de patrones GOF se utilizan de manera indispensable al enfrentarse a la programación. Estos patrones brindan las formas de crear instancias de objetos, describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades, además ayudan a definir la comunicación e iteración entre los objetos de un sistema, es decir reducen el acoplamiento entre los objetos.

- Estructurales: **Decorador**

Con el uso del Decorador se añade funcionalidad a una clase dinámicamente. El archivo *layout.html.twig* (o plantilla global) almacena el código HTML que es común a todas las páginas de la aplicación: *layout_admin.html.twig*, *layout_editor.html.twig*, *layout_profesor.html.twig* de los ambientes del Administrador, del Editor y del Profesor para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, lo cual implica que éste decora la plantilla.

- Comportamiento: **Observador**

El despachador de eventos de *Symfony2* implementa el patrón observador en una manera sencilla y efectiva para hacer todo esto posible y para hacer realmente extensibles tus proyectos. Se comportan como observadores que registran la observación de los principales eventos de *Symfony2*. Ejemplo de su utilización en el sistema es el archivo *LoginListener.php* para cuando los usuarios se registran.

- **Inyección de dependencia**

Una de las características más sobresaliente de *Symfony2* es el uso intensivo que hace de la Inyección de Dependencias, un potente patrón de diseño mediante el que se facilita la creación y configuración de objetos que prestan servicios en una aplicación gracias a la gestión automática de sus dependencias. Contribuye a crear un código más desacoplado y coherente.

Se comportan como un tipo de objetos que realizan algún tipo de tarea “global” en el sistema y se denominan *Servicios*, esto quiere decir que se crean y se destruyen en el momento en que han realizado su labor. Estos últimos objetos son más propios de la lógica de negocio de la aplicación. Ejemplo de su

utilización en el sistema es `services.yml` un servicio diseñado para registrar usuarios en el módulo de administración.

2.8 Descripción de la arquitectura

En la arquitectura propuesta que se muestra en la Figura 4 se utiliza uno de los estilos arquitectónicos existentes para facilitar y estandarizar su desarrollo. Se puede decir que se utilizará para la descripción de la arquitectura del sistema el **estilo** arquitectónico modelo-vista-controlador (MVC). Como **patrón** arquitectónico se utilizará el patrón modelo-vista-controlador ya que el estilo define los patrones posibles a utilizar en el desarrollo de las aplicaciones, además de mencionar que el mismo es el estilo que se utiliza en el desarrollo de las aplicaciones web. Por lo anteriormente dicho se puede afirmar que la arquitectura propuesta está separada en los componentes fundamentales que conforman el patrón arquitectónico, es decir las interfaces que se le muestran a los usuarios que son las llamadas vistas, las clases que conforman el modelo y las clases que controlan la lógica del sistema después de recibir la orden de los usuarios a través de la vista.

Se decidió el uso de este patrón arquitectónico pues MVC implica que el controlador reciba las llamadas de la vista, por lo tanto existe una dependencia de la vista al controlador, pero el controlador también conoce que vista se debe generar después de la siguiente como sucede en el caso de las aplicaciones web.

Para la realización de la arquitectura de SACOI-AA se tuvo en cuenta el diseño de una estructura física capaz de soportar el código, creando un esqueleto base definido por el framework a utilizar. Se basa en los estilos de llamada y retorno, fundamentalmente el referente al MVC, debido a que esta familia de estilos enfatiza la modificabilidad, la escalabilidad y la reusabilidad del sistema.

El sistema estará dividido en módulos, para lograr reusabilidad y facilitar el mantenimiento del mismo ante cambios que se puedan realizar en los procesos de negocio:

1. **Módulo Administración:** En gran parte, la seguridad del sistema se encuentra en la autenticación. Cada usuario posee un acceso de acuerdo al rol que desempeñe. Este módulo es el responsable de la autenticación y el registro en la aplicación.

- Módulo Diseño Instruccional:** En este módulo se encuentran las actividades que realizará el estudiante. Estas actividades son todas las propuestas por el profesor: encuestas, diagnósticos, ejercicios, conferencias, revisión bibliográfica. Además, le ofrece al estudiante el método de enseñanza que se corresponde a sus necesidades, creando una ruta de aprendizaje para cada estudiante que se encuentre registrado en el sistema.

Haciendo uso del MVC como arquitectura rigente para SACOI-AA, se realiza una distribución por capas de acuerdo a la misma:



Figura 4: Representación arquitectónica del sistema SACOI-AA en capas usando MVC.

Dónde:

Presentación (Vista): Conjunto de archivos que componen las páginas clientes del sistema. Contiene todo lo que es común para toda la aplicación: el *layout* (decora la plantilla de los módulos), los *.css* y archivos *.js* con la presentación *twit* de cada módulo.

Lógica de Negocio (Controlador): Conjunto de clases y archivos de configuración encargados de gestionar todos los procesos de la lógica de negocio. El controlador del sistema contiene los controladores frontales que son la vía de entrada y salida de la aplicación y los encargados de cargar todas las configuraciones de la aplicación, así como los de sus módulos. Los controladores de los módulos desarrollan toda la lógica del módulo.

Datos (Modelo): Conjunto de clases que controlan el tratamiento de los datos. *Doctrine2* contiene todas las clases de mapeo de la BD generada.

Base de Datos: Conjunto de tablas relacionadas que contienen los datos de la aplicación.

Diagramas de Clases

En la Figura 5 se muestra la representación del diagrama de clases del sistema. Se observa la relación entre las clases principales del sistema (usuario, rol, asignatura) y las relaciones con el resto de las clases:

Usuario: Esta clase almacena los datos principales de una persona registrada en el sistema. Estos datos cambian en dependencia del Rol que desempeñen los usuarios en determinada asignatura. Mantiene una relación de Muchos–Muchos con la clase Rol y una relación de Muchos–Muchos con la clase Asignatura. Un usuario puede tener varios roles y puede estar registrado en diferentes asignaturas.

Rol: Esta clase almacena los datos principales de un rol en dependencia del usuario registrado en una determinada asignatura del sistema. Los mismos son nombre del rol y su descripción. Mantiene una relación de Muchos–Muchos con la clase Asignatura. Un usuario puede tener varios roles como estudiante, profesor, profesor principal y administrador.

Asignatura: Esta clase almacena los datos principales de una asignatura en el sistema. Mantiene una relación fuerte con la clase Contenidos que define la cantidad de temas que tendrá la asignatura, una relación fuerte con la clase Objetivo y con la clase Habilidades ya que estas forman la parte más importante en una asignatura.

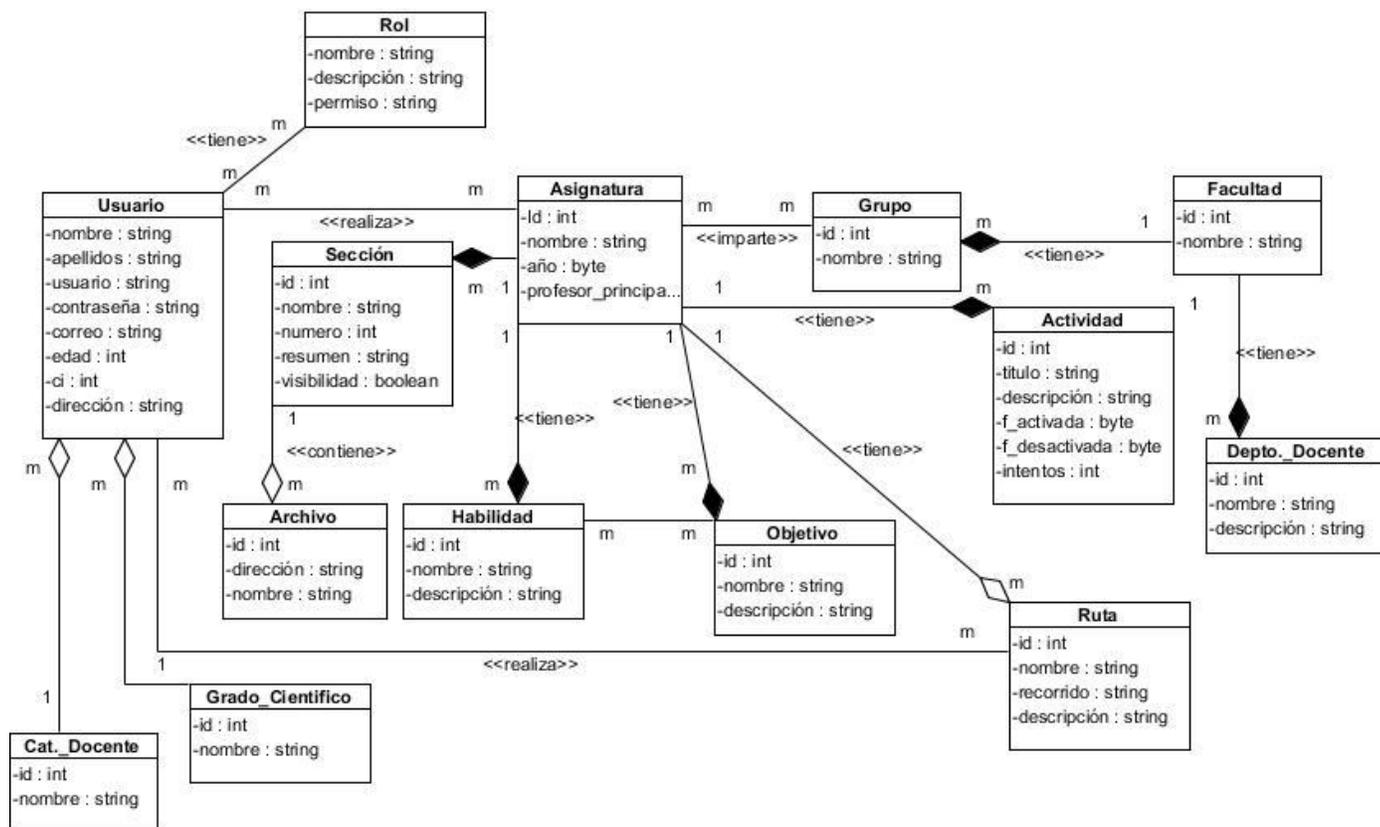


Figura 5: Representación del diagrama de clases.

2.9 Representación de los diagramas de Base de Datos

En la Figura 6 se representa el diagrama de base de datos correspondiente a la implementación de sistema SACOI-AA. Del mismo se hace una descripción de las tablas fundamentales.

Tabla Nom_Depto_Docente: lista de nombres determinados de los departamentos docentes de una facultad.

Tabla Nom_Cat_Docente: lista de nombres determinados de las categorías docentes.

Tabla Nom_Grado_Científico: lista de nombres determinados de los grados científicos.

Tabla Usuario: permite la gestión de los datos de un usuario en el sistema y mantiene una relación con la Tabla Rol y con la Tabla Asignatura.

Tabla Asignatura: Permite la gestión de los datos de una asignatura y mantiene relaciones con las Tablas Objetivo, Habilidades y Secciones por lo que permite realizar una asignación de estos a la asignatura.

Tabla Grupo: Tabla que permite realizar la asignación de estudiantes a un grupo en una determinada asignatura, ya que tiene una relación con esta tabla.

Tabla Facultad: Tabla que permite relacionar el grupo a una determinada facultad ya que mantiene una relación con la Tabla Grupo.

Tabla Ruta: Tabla que permite representar la ruta de aprendizaje que le será asignada al estudiante mediante la cual va a alcanzar el cumplimiento de los objetivos de la asignatura.

Tabla Actividad: Tabla que permite representar en la ruta de aprendizaje que le será asignada al estudiante las evaluaciones que tendrá que vencer para alcanzar el cumplimiento de los objetivos de la asignatura.

Tabla Archivo: Tabla que permite representar en la ruta de aprendizaje que le será asignada al estudiante los materiales mediante los cuales va a alcanzar el conocimiento necesario para luego someterse a las evaluaciones que le permitan cumplir los objetivos de la asignatura.

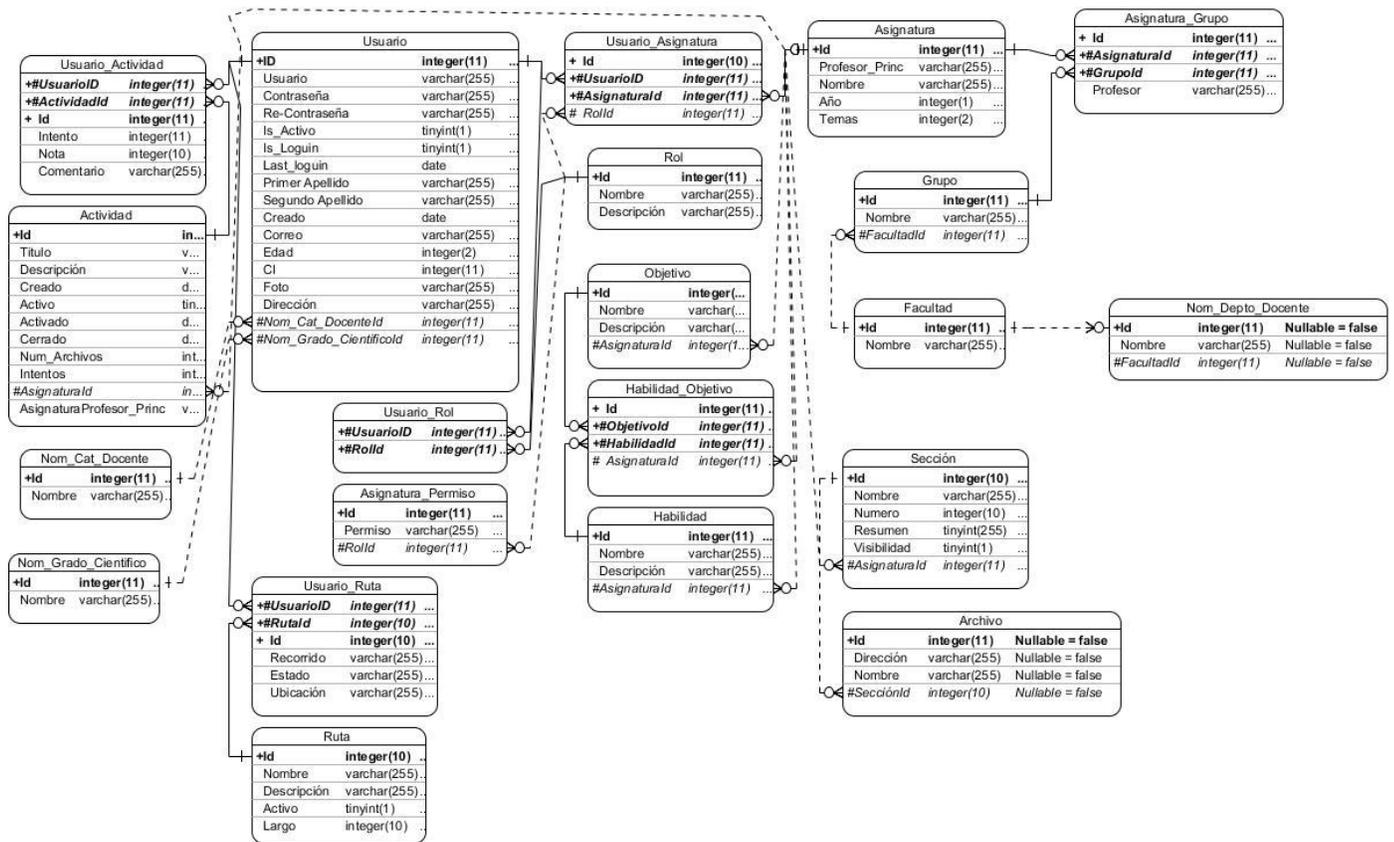


Figura 6: Representación del diagrama de Base de Datos.

En los anexos (Ver Anexos) se encuentran las Tarjetas CRC como otro de los artefactos generados por la metodología XP. Estas tarjetas brindan la posibilidad al programador de organizar su trabajo, ya que permiten representar la información más resumida de las responsabilidades de las clases de la base de datos. La información contenida en dichas tarjetas es:

- Nombre de la clase: clase a la que se le realiza la tarjeta CRC.
- Responsabilidades: funcionalidades que serán implementadas en esa clase.
- Colaboraciones: clases que permiten que sean llevadas a cabo las responsabilidades.

2.10 Conclusiones Parciales

Luego del análisis y diseño realizado se puede concluir lo siguiente:

- Se definió la arquitectura del sistema, se obtuvieron los artefactos correspondientes a la metodología que se seleccionó y la descripción de los mismos.
- Quedaron definidos 11 requisitos funcionales conformados en dos agrupaciones de requisitos. Se identificaron además ocho requisitos no funcionales con sus respectivas descripciones, todos objetos de informatización.
- Como parte de la validación de los requisitos, se definieron los prototipos de interfaz de usuario para el sistema que se será implementado, mostrándose estos en los Anexos.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

A partir de los resultados que se obtuvieron en el diseño, se comienza la implementación y las pruebas del sistema. En este capítulo se describe cómo realizar una exitosa implementación a partir del modelo del diseño en términos de clases. Se mostrará el diagrama de despliegue y para realizar la validación de la implementación se aplicarán pruebas unitarias (pruebas de caja blanca) y pruebas de aceptación con el cliente como parte de las pruebas de caja negra, además de realizar validación del método de expertos con la técnica Delphi, mediante la cual permitirán se validará teóricamente la pertinencia de la propuesta.

Fase Plan de lanzamiento

En esta fase el cliente establece la prioridad de cada historia de usuario. También se toman acuerdos sobre el contenido de cada entrega. A partir de la prioridad de las historias de usuario, se decide cuáles de ellas se implementarán en las primeras iteraciones.

3.1 Plan de iteraciones

Las HU seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada HU se traduce en tareas específicas de programación. Así mismo, para cada HU se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes. Por lo anteriormente mencionado se decide realizar la aplicación en dos iteraciones, las cuales se describen a continuación:

Iteración 1

En esta primera iteración se le dará cumplimiento a las historias de usuarios correspondientes a los requisitos funcionales 1, 2, 3, 4, 5, 6 y 7, las cuales serán de vital importancia para la aplicación, pues ellas conformarán la base de la estructura del sistema. Estas HU recogen las principales funcionalidades del sistema. Una vez que se terminen de implementar todas las funcionalidades de esta iteración, se procederá a liberar un ejecutable, el cual será mostrado al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo.

Iteración 2

En esta iteración se implementarán de las historias de usuarios restantes, las cuales están relacionadas a los requisitos funcionales: 8, 9,10, 11 y 12. Una vez que se termine de implementar estas funcionalidades se liberará la versión 1.0 de la aplicación.

3.2 Plan de duración de las iteraciones

Para una mayor organización del trabajo y como parte del ciclo de vida de un proyecto que utiliza la metodología XP se crea el plan de duración de cada una de las iteraciones Tabla 8, en este caso se hace para el único equipo de desarrollo con el cual se cuenta. Este plan tiene como objetivo mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada una de las mismas, lo que ayuda a obtener una idea aproximada del tiempo que durará la confección del sistema en su totalidad.

Iteración	Orden de las HU a implementar	Duración total de las iteración
1	Establecer los permisos correspondientes para la modificación y aprobación de los cambios a nivel de estudiante y profesor.	12 semanas.
	Adicionar un nuevo estudiante.	
	Modificar un estudiante.	
	Eliminar un estudiante.	
	Mostrar todos los estudiantes.	
	Adicionar un nuevo estudiante.	
	Modificar un estudiante.	
	Eliminar un estudiante.	
	Mostrar todos los estudiantes.	
	Adicionar una nueva asignatura.	
	Modificar una asignatura del sistema.	
	Eliminar una asignatura del sistema.	
	Mostrar todas asignaturas del sistema.	

2	Crear la ruta de aprendizaje típica de una asignatura.	8 semanas.
	Modificar la ruta de aprendizaje típica de una asignatura.	
	Eliminar la ruta de aprendizaje típica de una asignatura.	
	Mostrar todos los datos de la ruta de aprendizaje típica de una asignatura.	
	Crear cuestionarios, encuestas o autoevaluaciones de acuerdo a las habilidades y que verifiquen el grado de cumplimiento de cada objetivo instructivo de la asignatura.	
	Graficar la ruta a seguir según el nivel de asimilación de cada estudiante por cada contenido de la asignatura de acuerdo al tiempo entre una actividad docente y la siguiente.	
	Emitir un reporte del nivel de completamiento de los objetivos instructivos de la asignatura de acuerdo a los contenidos vencidos y a las habilidades logradas (clases prácticas resueltas, autoevaluaciones realizadas).	

Tabla 8: Distribución de historias de usuario por iteración.

3.3 Plan de entregas

Como propuesta del plan de entregas se harán versiones al sistema en las fechas aproximadas que se indican a continuación:

Producto:

1. Final 1ra Iteración 3^{ra} semana de Abril.
2. Final 2da Iteración 3^{ra} semana de Mayo.

3.4 Representación del Diagrama de Despliegue

A continuación en la Figura 7 se realiza una descripción de los nodos y protocolos del diagrama de despliegue.

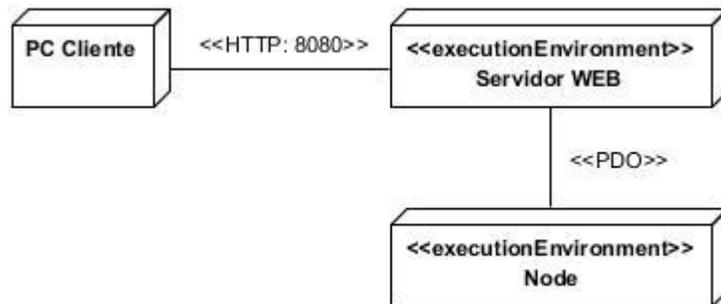


Figura 7: Representación del diagrama de despliegue.

Descripción de los nodos

- Cliente: representa una computadora desde la cual el usuario podrá acceder a la aplicación.
- Servidor Web: representa una estación donde estará montado el servidor Apache sobre el cual se estará ejecutando la aplicación.
- Servidor de BD: representa el servidor donde estará el sistema gestor de base de datos PostgreSQL que dará respuesta a las peticiones hechas por la aplicación.

Descripción de los Protocolos

- HTTP: establece un esquema de comunicación cliente/servidor. El cliente es el navegador web que realiza las peticiones a las que el servidor se encarga de dar respuesta.

Fase Producción y Validación

3.5 Diseño y ejecución de las Pruebas de Software

Una de las etapas fundamentales de la metodología XP la constituye el proceso de pruebas, el cual anima a los desarrolladores a probar lo que hacen constantemente. Mediante esta filosofía se minimiza la cantidad de errores no detectados y también el tiempo transcurrido entre la introducción de dichos errores en la aplicación y su detección. Todo esto contribuye a elevar la calidad del producto y a la seguridad de los desarrolladores a la hora de hacer cambios o modificaciones. Una de las características de la metodología XP, es que divide las pruebas en dos grupos. Seguidamente se explica en detalles sus particularidades.

3.6 Validación mediante pruebas unitarias

Pruebas unitarias: Las pruebas unitarias se establecen antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema, se basa en hacer pruebas a pequeños fragmentos de código encargados de una tarea específica. Generalmente son desarrolladas por los programadores y encargadas de verificar el código de forma automática.

El objetivo de realizar este tipo de prueba, es demostrar a través de inspecciones al código que este no contiene errores y en caso de contener proceder a su pronta eliminación, asegurando que cada módulo implementado funcione adecuadamente. Estas pruebas al sistema verifican que el mismo se comporta de la manera esperada, por lo que podrían encajar dentro de la definición de pruebas unitarias que propone XP. La biblioteca de pruebas PHPUnit está en el centro de las pruebas de Symfony2. Cuando se ejecuta PHPUnit este buscará en los directorios del sistema las pruebas a ejecutar.

Pruebas de Caja Blanca

Este tipo de pruebas de software se le realiza a funciones internas ó a los métodos de las clases pertenecientes a un módulo-componente del sistema Figura 8. Se basan en el minucioso examen de los procedimientos del código a evaluar, por lo que se hace necesario conocer la lógica del programa. Se comprueban los caminos lógicos del software, proponiendo casos de prueba que examinen que están correctas todas las condiciones. Entre las técnicas usadas se encuentran la cobertura de caminos ciclómicos, es decir las que hacen que se recorran todos los posibles caminos de ejecución.

Características Fundamentales

- Garantizan que se ejecuten al menos una vez todos los caminos, independientemente de cada módulo.
- Ejercitan todas las decisiones lógicas por sus variantes verdadera o falsa.
- Ejecutan todos los ciclos.
- Ejercitan las estructuras internas de datos para asegurar su validez.

Para aplicar esta técnica se siguen los siguientes pasos:

1. Dibujar el grafo asociado a partir del diseño o código fuente.
2. Calcular la complejidad ciclómica del grafo.

3. Determinar un conjunto básico de caminos independientes.
4. Preparar los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

```

public function adicionarRutaAction($id, $id_ruta){

    $em      = $this->getDoctrine()->getManager();           -----//1
    $ruta    = $em->getRepository("AdminBundle:Ruta")->find($id_ruta);-----//1

    $entity  = $ruta->getAsignatura();-----//1
    $section = $em->getRepository('AdminBundle:Asignatura')->findSections($entity->getId());-----//1

    $nodes = array();-----//1
    if($ruta){-----//2
        $json_ruta = $ruta->getRuta();-----//3
        $array_ruta = json_decode($json_ruta, true);-----//3

        if(!$array_ruta){-----//4
            $array_ruta = null;-----//5
        }else{
            foreach($array_ruta as $value){-----//6
                $node_source_name = $value['source'];-----//6
                $node_target_name = $value['target'];-----//6
                $node_pathS = $this->container->get("node.". $value['sourceT']);-----//6
                $node_pathT = $this->container->get("node.". $value['targetT']);-----//6

                $ubicacion = explode("-", $node_source_name);-----//6
                $ubicacion_id = $ubicacion[1];-----//6
                $entityS = $node_pathS->find($ubicacion_id, $em);-----//6
                $nameS = $node_pathS->getNodeName($entityS);-----//6

                $ubicacion = explode("-", $node_target_name);-----//6
                $ubicacion_id = $ubicacion[1];-----//6
                $entityT = $node_pathT->find($ubicacion_id, $em);-----//6
                $nameT = $node_pathT->getNodeName($entityT);-----//6
            }
        }
    }
}

```

```

        $nodes[$node_source_name] = array('id'=>$node_source_name,'position'=>$value['sourceP'],
                                          'type'=>$value['sourceT'],'control'=>$node_pathS->isNodeControl(),
                                          'name'=>$node_pathS->getType().".".$nameS);--//6
        $nodes[$node_target_name] = array('id'=>$node_target_name,'position'=>$value['targetP'],
                                          'type'=>$value['targetT'],'control'=>$node_pathT->isNodeControl(),
                                          'name'=>$node_pathT->getType().".".$nameT);--//6
    }
}
}
return $this->render('AdminBundle:Ruta:nuevaruta.html.twig',array(
    'entity'      => $entity,
    'secciones'   => $section,
    'ruta_entity' => $ruta,
    'ruta'        => $array_ruta,
    'asig_id'     => $id,
    'nodes'       => $nodes));-----//7

```

Figura 8: Representación del Método AdicionarRutaAction de la clase RutaController.

Al término de la identificación de cada línea de código, es necesario representar el grafo de flujo asociado Figura 9, en el cual se representan los distintos puntos que lo componen. En el mismo se pueden identificar los siguientes elementos:

Nodo: Son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión.

Nodo de predicado: Son los nodos que contienen una condición y se caracterizan porque de ellos salen dos o más aristas.

Aristas: Son constituidas por las flechas del grafo, estas terminan en un nodo.

Regiones: Son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento.

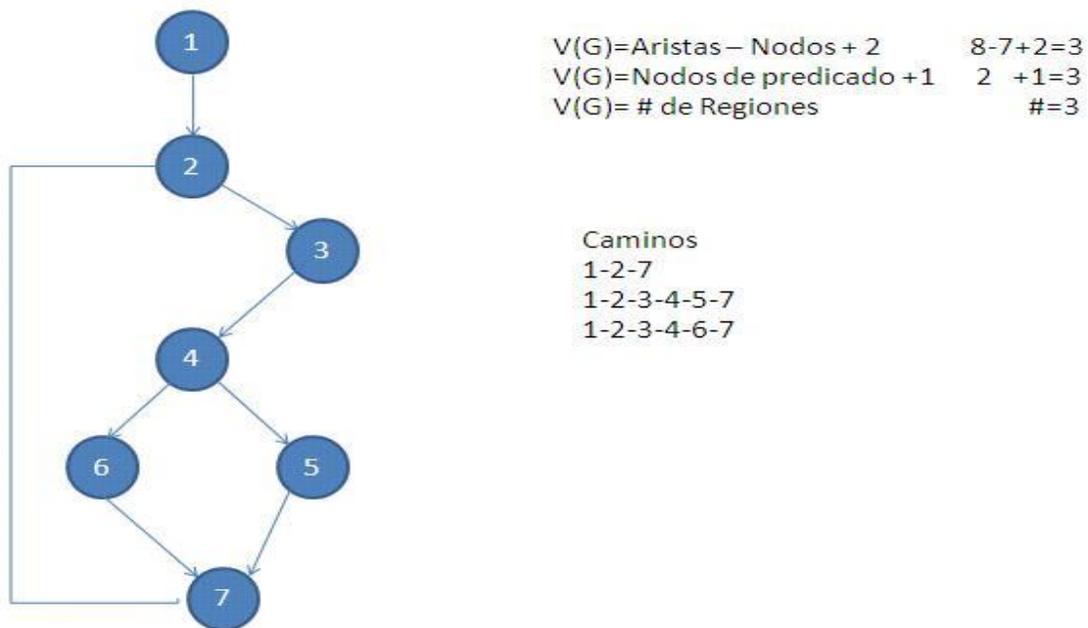


Figura 9: Grafo de Flujo asociado al Método AdicionarRutaAction ().

La complejidad ciclomática coincide con el número de regiones de un grafo de flujo.

La complejidad ciclomática, $V(G)$, de un grafo de flujo G , se define como $V(G) = \text{Aristas} - \text{Nodos} + 2$.

La complejidad ciclomática, $V(G)$, de un grafo de flujo G , también se define como $V(G) = \text{Nodos de predicado} + 1$.

A partir del grafo de flujo de la Figura 9, la complejidad ciclomática sería:

Como el grafo tiene tres regiones, **$V(G) = 3$**

Como el grafo tiene 8 aristas y 7 nodos, **$V(G) = 8 - 7 + 2 = 3$**

Como el grafo tiene 2 nodos predicado, **$V(G) = 2 + 1 = 3$**

Teniendo la complejidad ciclomática se obtiene además el número de caminos independientes, que ofrecen un valor límite para el número de pruebas que se deben diseñar. En el ejemplo mostrado en la Figura 9, el número de caminos independientes es 3.

Para cada camino se realiza un caso de prueba

No. De caminos	Casos de prueba	Objetivo	Resultados esperados
1	adicionarRutaAction(\$id, \$id_ruta)	Si existe esta ruta.	Se muestra la ruta.
2	adicionarRutaAction(\$id, \$id_ruta)	Si no hay archivos para crear la ruta.	Se muestra la misma página pero en blanco.
3	adicionarRutaAction(\$id, \$id_ruta)	Si hay archivos para crear la ruta.	Se muestra la misma página pero con la ruta que se crea.

Al finalizar la confección de los casos de prueba descritos anteriormente y de aplicados estos al algoritmo que se le realizó la prueba de caja blanca, el equipo de desarrollo determinó que el flujo de trabajo de dicho algoritmo es correcto pues cumplió con las condiciones necesarias de la prueba.

3.7 Validación mediante pruebas de aceptación

Las pruebas de aceptación se crean a partir de las historias de usuario. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de estas pruebas es garantizar que los requerimientos hayan sido cumplidos y que la aplicación es realmente lo que el cliente quería. Una HU no se considera terminada hasta que no haya pasado sus pruebas de aceptación.

Por otro lado las pruebas de aceptación están destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que sea la funcionalidad esperada por el cliente. Tienen como objetivo verificar la funcionalidad del sistema a través de sus interfaces externas comprobando que dicha funcionalidad sea la esperada en función de los requisitos del sistema.

Las pruebas de aceptación son tan importantes como las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado al final de una iteración. Por esta razón se decide utilizar como uno de los métodos de validación del sistema este tipo de pruebas para garantizar el buen

funcionamiento de la aplicación y así la satisfacción del cliente con el desarrollo del mismo. Seguidamente se muestran (Ver Tabla 9, Tabla 10, Tabla 11, Tabla 12, Tabla 13) las pruebas de aceptación del RF6 “Gestionar el diseño Instruccional de una asignatura”, perteneciente al módulo de Diseño Instruccional correspondiente a la 2da iteración.

Tabla 9: Caso de prueba de aceptación: HU10.

Casos de Prueba	
Número de caso de prueba :CP_1	Número de Historia de Usuario:HU27
Nombre del Caso de prueba: Adicionar la ruta de aprendizaje de una asignatura.	
Condiciones de ejecución: El usuario debe estar autenticado y tener permisos para realizar estas operaciones, es decir debe tener rol de profesor principal o Editor.	
Escenario de prueba 1	
Entradas: 1-Ir a la sección Ruta de aprendizaje. 2-Hacer clic en la opción Nueva.	
Resultado esperado: El sistema muestra la opción para llenar los datos de la nueva ruta de aprendizaje.	
Escenario de prueba 2	
Entradas: Introducción correcta de los datos.	
Resultado esperado: Se muestra un mensaje indicando que la ruta fue creada con éxito.	
Escenario de prueba 3	
Entradas: No introducir los datos.	
Resultado esperado: Se muestra un mensaje indicando que alguno de los campos está vacío, permitiendo introducirlo de nuevo.	
Evaluación: Satisfactoria.	

Tabla 10: Caso de prueba de aceptación: HU11.

Casos de Prueba	
Número de caso de prueba :CP_2	Número de Historia de Usuario:HU28
Nombre del Caso de prueba: Mostrar la ruta de aprendizaje de una asignatura.	
Condiciones de ejecución: El usuario debe estar autenticado y tener permisos para realizar estas operaciones, es decir debe tener rol profesor principal o Editor.	
Escenario de prueba 1	
Entradas: 1-Ir a la sección Ruta de aprendizaje. 2-Hacer clic en la opción Listar todos.	
Resultado esperado: Se muestran todas las rutas adicionadas en el sistema.	
Escenario de prueba 2	
Entradas: 1-Ir a la Ruta de aprendizaje que desea conocer sus datos 2-Hacer clic en la opción Ver.	
Resultado esperado: Se muestra toda la información perteneciente a la ruta de aprendizaje seleccionada.	

Evaluación: Satisfactoria.

Tabla 11: Caso de prueba de aceptación: HU12.

Casos de Prueba	
Número de caso de prueba :CP_3	Número de Historia de Usuario:HU29
Nombre del Caso de prueba: Modificar la ruta de aprendizaje de una asignatura.	
Condiciones de ejecución: El usuario debe estar autenticado y tener permisos para realizar estas operaciones, es decir debe tener rol de administrador.	
Escenario de prueba 1	
Entradas: 1-Ir a la sección Ruta de aprendizaje. 2-Hacer clic en la opción Listar todos.	
Resultado esperado: El sistema muestra todas las rutas adicionadas en el sistema.	
Escenario de prueba 2	
Entradas: 1-Ir a la Ruta de aprendizaje que desea modificar sus datos 2-Hacer clic en la opción Editar Datos.	
Resultado esperado: Se muestra toda la información perteneciente a la ruta de aprendizaje seleccionada y esta información puede ser modificada.	
Escenario de prueba 3	
Entradas: El administrador introduce datos incorrectos.	
Resultado esperado: Se muestra un mensaje indicando que alguno de los datos es incorrecto, permitiendo introducirlo de nuevo.	
Escenario de prueba 4	
Entradas: El administrador deja campos vacíos.	
Resultado esperado: Se muestra un mensaje indicando que alguno de los campos está vacío, permitiendo introducirlo de nuevo.	
Evaluación: Satisfactoria.	

Tabla 12: Caso de prueba de aceptación: HU13.

Casos de Prueba	
Número de caso de prueba :CP_4	Número de Historia de Usuario:HU30
Nombre del Caso de prueba: Eliminar la ruta de aprendizaje de una asignatura.	
Condiciones de ejecución: El usuario debe estar autenticado y tener permisos para realizar estas operaciones, es decir debe tener rol profesor principal o Editor.	
Escenario de prueba 1	
Entradas: Selección de la ruta a eliminar.	
Resultado esperado: La ruta es eliminada correctamente.	
Escenario de prueba 2	
Entradas: Selección de la ruta a eliminar y esta contiene estudiantes asignados	
Resultado esperado: La ruta no puede ser eliminada por tener estudiantes asignados a ella.	

Evaluación: Satisfactoria.

Tabla 13: Caso de prueba de aceptación: HU14.

Casos de Prueba	
Número de caso de prueba :CP_5	Número de Historia de Usuario:HU31
Nombre del Caso de prueba: Graficar la Ruta de aprendizaje a seguir por cada estudiante.	
Condiciones de ejecución: El usuario debe estar autenticado y tener permisos para realizar estas operaciones, es decir debe tener rol de profesor principal o Editor.	
Escenario de prueba 1	
Entradas: 1-Ir a la sección Ruta de aprendizaje. 2-Hacer clic en la opción Listar todos.	
Resultado esperado: El sistema muestra todas las rutas adicionadas en el sistema.	
Escenario de prueba 2	
Entradas: 1-Ir a la Ruta de aprendizaje que desea crear. 2-Hacer clic en la opción Crear Ruta.	
Resultado esperado: Se muestra un cuadro vacío hacia donde se pueden arrastrar los elementos que conformaran la Ruta de aprendizaje, además de una columna con toda la documentación necesaria por temas, para conformar la ruta de aprendizaje.	
Evaluación: Satisfactoria.	

3.8 Validación mediante Técnica Delphi del criterio de expertos

Para validar teóricamente la idea a defender de la investigación se propone utilizar el método de experto en su variante Delphi, con el objetivo de elaborar pronósticos a largo plazo, teniendo en cuenta la utilización sistemática de las valoraciones intuitivas de un grupo de expertos para obtener un consenso de opiniones que de perfecto acuerdo entre las partes, refuerzan la validez de la propuesta.

El primer paso que se siguió con este método fue la selección de un grupo de expertos. Durante este período, ningún experto conocerá la identidad de los otros miembros del grupo de debate. Las etapas propuestas para la realización del método son las siguientes:

1. Selección de expertos. Estudio preliminar de expertos para la toma de decisiones.
2. Elaboración del cuestionario, para la validación de SACOI-AA.
3. Determinación de la concordancia de los expertos.
4. Desarrollo práctico y explotación de los resultados.

Selección de los expertos

Se considera experto a la persona, grupo de personas u organización con amplio conocimiento en la temática, gran experiencia y calificación. La selección se realizó teniendo en cuenta los siguientes criterios:

1. Graduado del nivel superior: Deben haber culminado los estudios en el nivel superior. Adquirido y desarrollado habilidades para explicar y valorar fenómenos, argumentar, demostrar y hacer resúmenes, lo que posibilita obtener un mayor desarrollo para el área de conocimiento en que se desenvuelve.
2. Tres años de experiencia como mínimo en el trabajo de los temas de Gestión Docente: Se exigirá que las personas en condición de experto deben tener experiencia en el trabajo con la Gestión Docente. Así garantizan dominio en el tema y por lo tanto resultan claves para emitir criterios relacionados con el modelo de que se presenta.
3. Prestigio en el colectivo de trabajo: Deben poseer prestigio en el colectivo de trabajo, dado por la participación en eventos, publicaciones, investigaciones relacionadas con el tema, interés en la temática desarrollada, poder de decisión, años de experiencia, categoría científica, experiencias prácticas, etc.

Seguidamente se le realizó una encuesta de autoevaluación a cada uno de los posibles expertos para determinar el coeficiente de competencia Ver Anexo 5.

Cálculo del Coeficiente de Competencia

Para la selección de los expertos se tuvo en cuenta la valoración de competencias, por lo que se calculó el coeficiente de competencia (k) a través de la siguiente fórmula:

$$K = \frac{1}{2}(Kc + Ka)$$

Kc: es el coeficiente de conocimientos.

Ka: es el coeficiente de argumentación.

Dónde:

Los valores del coeficiente de conocimiento se obtienen de la pregunta 1 de la encuesta. En la (Ver **Tabla 14**) se muestran los resultados.

Cálculo del coeficiente de conocimiento

Tabla 14: Cálculo del coeficiente de conocimiento

1	2	3	4	5	6	7	8	9	10
								X	

El experto debe marcar en la casilla enumerada en una escala del 1 al 10, según su criterio sobre la capacidad que tiene en el tema. Luego se ajusta a la teoría de las probabilidades y se multiplica por 0.1. La evaluación “0” indica que el experto no tiene conocimiento de la problemática correspondiente. Por ejemplo, el coeficiente de conocimientos (**Kc**) del experto 9 es 0.95 (Alto), pues su selección se centra en la casilla número nueve.

Para calcular el coeficiente de argumentación se procede de la siguiente forma: el experto debe marcar en este caso los elementos que le permiten argumentar su evaluación del nivel de conocimiento seleccionado en la pregunta 1, (Ver Anexos).

De acuerdo a los valores que aparecen en la tabla Patrón para el Cálculo del coeficiente de argumentación, y según las casillas marcadas por cada encuestado, se calcula el coeficiente de argumentación.

Entonces el coeficiente de argumentación sería: **Ka = 0,3 + 0,5 + 4(0,05)= 1**

La selección final de los expertos tendrá lugar atendiendo a los siguientes criterios de interpretación del coeficiente de competencia:

(K): $0.5*(Ka + Kc) 0.5*(1+0.9) = 0.95$

Para interpretar el coeficiente de competencia se utiliza la siguiente escala:

Si **$0.8 \leq k \leq 1.0$** , el coeficiente de competencia es alto.

Si **$0.5 \leq k < 0.8$** , el coeficiente de competencia es medio.

Si **$k < 0.5$** , el coeficiente de competencia es bajo.

Los expertos seleccionados para la validación fueron aquellos cuyos resultados mostraron un coeficiente de competencia alto y medio. Se concentran en la UCI, todos con amplio conocimiento en temas relacionados con la gestión docente (Ver Anexos). Se caracterizaron según su categoría docente, grado científico, años de experiencia en la temática de Gestión docente y como docentes.

Elaboración y lanzamiento de los cuestionarios

En este momento se elabora un cuestionario que se adapte a las características de los expertos para validar el sistema SACOI-AA que se propone.

Definición de los atributos

Para realizar la evaluación del modelo se definen los indicadores o atributos que serán evaluados por los expertos. A partir de estos se confecciona un cuestionario para que los expertos expresen su valoración en relación con el modelo. Los atributos que se tuvieron en cuenta fueron los siguientes:

A1: Valor Científico.

P1-Necesidad de implementación de un sistema que permita mejorar el uso que se le da a la información y los datos que se obtienen de los estudiantes durante el proceso de enseñanza-aprendizaje.

A2: Operatividad.

P2, P5, P7-Coherencia en la estructura modular del sistema.

A3: Facilidad de uso.

P3, P4, P6- Necesidad de que el sistema pueda ser utilizado por usuarios no necesariamente expertos en informático.

A4: Generalidad del sistema propuesto.

P8- Independencia del sistema a entornos universitarios específicos.

Determinación de la concordancia de los expertos

El Coeficiente de Concordancia de Kendall, constituye un estadígrafo muy útil en estudios de confiabilidad entre expertos de una materia. Es una medida de coincidencia entre ordenaciones que pueden ser objetos o individuos. La fórmula planteada por Kendall para calcular el Coeficiente de Concordancia es la siguiente:

$$W = \frac{K * S}{K^2(N^3 - N)}$$

N: Número de entidades (objetos, individuos) ordenados, es decir cantidad de aspectos a evaluar por los expertos.

Š: Suma de los rangos divididos entre la cantidad de preguntas realizadas.

K: Cantidad de expertos seleccionados para la validación.

W: Concordancia entre los expertos.

Dónde:

S: es la suma de los cuadrados de las desviaciones observadas de la media de S_j (rangos):

Para la aplicación del Coeficiente de Concordancia de Kendall (W), se elaboró la tabla (Ver Anexos) con los aspectos a evaluar contra expertos obteniéndose los rangos de valoración (S_j). En la Tabla 15 se muestra la correspondencia entre las posibles respuestas dadas por los expertos sobre el cuestionario y los valores asignados.

Tabla 15. Rango de los valores para el cálculo del coeficiente de Kendall. Fuente: Elaboración propia

Valor 1	No necesaria	No adecuada	Difícil	No vinculada
Valor 2	Poco necesaria	Poco Adecuada	Poco fácil	Poco vinculada
Valor 3	Necesaria	Adecuada	Fácil	Vinculada
Valor 4	Muy necesaria	Muy adecuada	Muy fácil	Muy vinculada

En dependencia de la evaluación que otorgue el experto a cada pregunta será el valor asociado que se asignará. El resultado de las evaluaciones se convierte en valores para obtener los rangos.

Los datos para realizar el cálculo de **W**, se muestran a continuación:

Cálculo de W:

$$\bar{S} = \frac{\sum_{j=1}^n S_j}{N}$$

Donde **N**= 6

Por lo tanto, quedaría:

$$\hat{S}=18.56$$

$$S_j=14,9$$

Luego $K=6$, $N=8$, para lo cual: $W= 0.0098$ expresa el grado de concordancia entre los diez expertos al dar un orden evaluativo a los aspectos valorados. Este coeficiente siempre será positivo y su valor estará comprendido en el rango de 0 a 1

Cálculo del Chi Cuadrado Real:

Si $X^2 \text{ real} < X^2 (\alpha, N - 1)$ entonces existe concordancia entre los expertos. El Chi cuadrado calculado se compara con los valores de las tablas estadísticas probadas para tales efectos con $\alpha=0,05$ y $N= 6$, para un nivel de confianza del 95 %.

Se busca en la tabla de distribución estadística $X^2 (0,05; 5)= 11,0705$ comprobándose que:

$$X^2 \text{ real} < X^2 (\alpha, N - 1)$$

$0.4132 < 11,0705$ La comparación realizada entre el valor de X^2 real y el valor tabulado, permite concluir que existe concordancia entre los expertos consultados.

Desarrollo práctico y exploración de los resultados

Los expertos que conformaron el panel recibieron un resumen del sistema de Gestión Docente SACOI-AA como documentación para responder los temas encuestados, además del cuestionario. A continuación se explica cómo se llegó a los resultados que se exponen en cada uno de los pasos del procesamiento de las encuestas. Se confeccionaron tablas para ir recogiendo los resultados aportados por los expertos y como herramienta se utilizó Microsoft Excel 2010 para procesar los cálculos de manera automática. Los resultados se recogieron en una tabla de doble entrada, preguntas (ocho) contra valor (rangos de los valores para el cálculo del coeficiente de Kendall) para el cálculo de la frecuencia absoluta, frecuencia absoluta acumulada y frecuencia relativa acumulada (Ver Anexos).

En la Figura 10 se muestran los puntos de corte utilizados para determinar la categoría o grado de adecuación de cada criterio según la opinión de los expertos consultados. Los rangos son los siguientes:

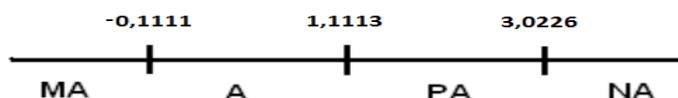


Figura 10: Rangos obtenidos a partir de los puntos de corte. Fuente: Elaboración propia.

MA: Muy Aceptable. **A:** Aceptable. **PA:** Poco Aceptable. **NA:** No Aceptable.

✓ **Resultados de la validación del modelo**

En la validación del modelo de Gestión Docente participaron nueve expertos con experiencia en la producción de software y la Gestión Docente respectivamente. De ellos 1 ingeniero y 8 máster. Todos

fueron seleccionados pues tenían un coeficiente de competencia alto y medio. En la Figura 11 se muestra el resumen de los resultados obtenidos de la encuesta de autovaloración del nivel de competencia de cada uno de los encuestados:

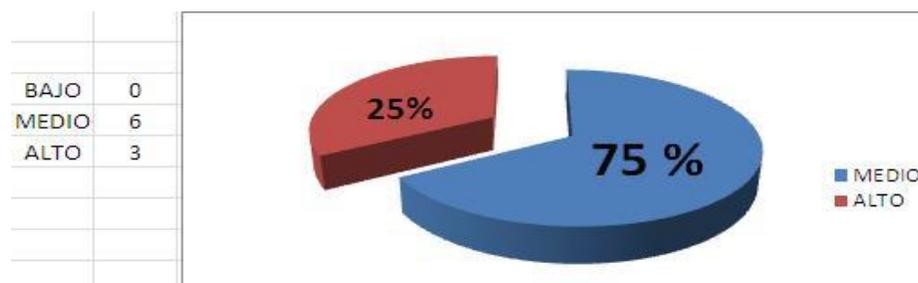


Figura 11. Resultados de las encuestas de autoevaluación de expertos.

De las preguntas formuladas en el cuestionario de evaluación el número dos y tres dedicadas a la adecuada estructura y facilidad de comprensión del modelo respectivamente fueron las que obtuvieron menor puntuación por parte de los expertos. Sin embargo al recomendarse los elementos puntuales que debían considerarse se corrigió la propuesta para que tuviera mayor aceptación y se revisó en el acto con los expertos que habían notado tal dificultad, quedando aceptado el modelo que se propone.

En la Figura 12 se representa el resultado final de la evaluación realizada al modelo por los expertos, donde los aspectos sometidos a consideración fueron evaluados de “Muy Aceptable MA” y de “Aceptable A”, demostrando el grado de aceptación de la propuesta.

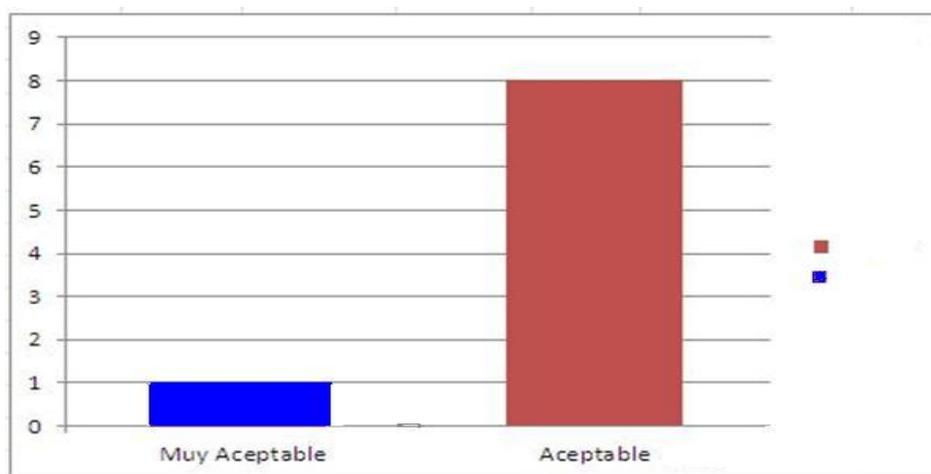


Figura 12: Categoría otorgada por los expertos al sistema SACOI-AA.

Luego de realizar el procesamiento de los resultados de la encuesta y tener el nivel de adecuación de cada una de las preguntas para estimar el grado de aceptación de la propuesta, se considera que se cumplieron con los objetivos propuestos, pues todas las preguntas obtuvieron nivel de adecuación “Aceptable”, por lo tanto puede considerarse por concluida la validación teórica del sistema de Gestión Docente SACOI-AA.

3.9 Conclusiones Parciales

Luego de desarrollada la implementación y validación del sistema se puede concluir que:

- Mediante las pruebas de aceptación el cliente mostró conformidad con el sistema presentado, ya que las funcionalidades estaban en correspondencia con lo planificado en la investigación.
- Las pruebas de caja de blanca permitieron detectar posibles errores cometidos en el sistema durante su desarrollo, los cuales fueron corregidos en breve.
- Con la aplicación del método experto se validó la pertinencia teórica del sistema SACOI-AA.
- Se realizó el plan de entrega y el diagrama de despliegue que permite conocer la necesidad de recursos para la implantación del sistema.

CONCLUSIONES GENERALES

Sobre el estudio realizado, se concluye lo siguiente:

- Se formalizó el marco teórico conceptual de la investigación a partir de un estudio realizado de las principales técnicas de inteligencia artificial y específicamente de las Analíticas de Aprendizaje en la Educación Superior, dando cumplimiento de este modo al primer objetivo específico.
- Se determinaron las herramientas que permitieron la realización de un sistema para el análisis del cumplimiento de objetivos instructivos utilizando técnicas de Analíticas de Aprendizaje y dando cumplimiento de este modo al segundo objetivo específico.
- Se realizaron pruebas de aceptación para validar que las funcionalidades implementadas coincidieran con las necesidades del cliente. Por último como método para realizar la validación de la idea a defender se utilizó la técnica Delphi del método de expertos, cumpliendo de esta manera con el tercer objetivo definido.

Sobre el sistema propuesto se concluye lo siguiente:

- El sistema permite a los profesores comprobar el cumplimiento por parte de sus estudiantes de los objetivos instructivos en la asignatura.
- El sistema muestra al estudiante la ruta de aprendizaje que debe seguir para el cumplimiento de los objetivos en una asignatura.
- Genera reportes con información sobre el proceso docente seguido por los estudiantes, lo cual constituye retroalimentación del proceso lectivo en la asignatura para el profesor.

RECOMENDACIONES

Para investigaciones futuras se recomienda incorporar las siguientes funcionalidades al sistema:

- Lograr la interoperabilidad de SACOI con los sistemas que integran la nube computacional de la UCI.
- Implementar una de las técnicas de RBC que permita un razonamiento autónomo e inteligente al sistema.

BIBLIOGRAFÍA

- 1 Álvarez Zayas, C. M. (1996) "La Universidad como Institución."
- 2 Arencibia Ramirez, J. (2011). Predicción de resistencia a fármacos del VIH utilizando multclasificadores. Centro de Identificación y Seguridad Digital. La Habana,, UNiversidad de las Ciencias Informaticas. Master: 80.
- 3 Ávila Mejía, O. (2011) "Computación en la nube.", pp. 45-52.
- 4 Barrón Estrada, M. L. (2011). Herramienta de autor para la identificación de estilos de aprendizaje. Revista Electrónica de Investigación Educativa. <http://redie.uabc.mx/vol13no1/contenido-zatarainbarron.html>. 13: 13.
- 5 Bonet Cruz, I. (2008). Modelo para la clasificación de secuencias, en problemas de la bioinformática, usando técnicas de inteligencia artificial. CIENCIA DE LA COMPUTACIÓN. Santa Clara,, UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS. Master: 130.
- 6 Brito Acuña, K. (2009). Selección de Metodologías de Desarrollo para Aplicaciones Web en la Facultad de Informática de la Universidad de Cienfuegos. Andalucía. 13: 240.
- 7 C. Berliner, D. (2006). Handbook of Educational Psychology: A Project of Division 15, the Division of Educational Psychology of the American Psychological Association. A. Books.
- 8 Canales Mora, R. (2009) "Patrones Arquitectonicos."
- 9 Castañeda Hevia, Á. E. (2003). El papel de las tecnologías de la informacion y las comunicaciones(TIC) en el proceso de enseñanza-aprendizaje a comienzos del siglo XXI. Preparación pedagógica integral para profesores universitarios
- 10 M. E. d. I. V. Garcia. La Habana, Felix Varela: 317.
- 11 Castillo, E. (1996). Sistemas Expertos y Modelos de Redes Probabilísticas. Cantabria, Universidad de Cantabria.
- 12 Cusumano, M. (2010). 5 Puntos de vistas. Technology Strategy and Management. http://delivery.acm.org/10.1145/1730000/1721667/p27-cusumano.pdf?ip=200.55.140.181&acc=ACTIVE%20SERVICE&CFID=144550267&CFTOKEN=51191340&acm_=1353433603_58e28ee7b978ac7a6bd74ea09ea1df4e, communications of the acm. 53: 3.
- 13 Daniel, J. (2005). Las tecnologías de la información y la comunicación en la formación docente. Montevideo.
- 14 Dikaiakos, M. D. a. P., G. (2009) "Cloud Computing. IEEE Computer Society." IEEE Computer Society, pp. 1-4.

- 15 Duarte Correa, Y. (2010). Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana, Universidad de las Ciencias Informáticas: 79.
- 16 Ecurrred (2010). Nuevas Tecnologías para la Educación. Ecurrred. La habana, Equipo de redacción.
- 17 Eguiluz, J. (2011). Desarrollo web ágil con Symfony2. Madrid.
- 18 Esteve, F. (2009). Bolonia y las TIC: de la docencia 1.0 al aprendizaje 2.0
- 19 Bolonia y las TIC: de la docencia 1.0 al aprendizaje 2.0. <http://www.francescesteve.es>. 5: 59-68.
- 20 Ferro Soto, C. (2009). Ventajas del uso de las TIC en el proceso de enseñanza-aprendizaje desde la óptica de los docentes universitarios. EDUTEC Revista Electrónica de Tecnología Educativa. http://edutec.rediris.es/Revelec2/revelec29/articulos_n29_pdf/5Edutec-E_Ferro-Martinez-Otero_n29.pdf: 12.
- 21 Gaceta Oficial, R. d. C. (2013) "Reestructuran Ministerio de Informática y las Comunicaciones." Cubadebate, 1.
- 22 Gálvez Lio, D. (2006). Sistemas basados en conocimientos. Universidad Central de Villa Clara Marta Abreu: 66.
- 23 García Lorenzo, M. M. (2012). Sistemas Basados en el Conocimiento. Villa Clara: 66.
- 24 Gosling, J. (2012). "The Java Language Specification " 2012.
- 25 Gross Salvat, B. (1992) "http://net.educause.edu/ir/library/pdf/ecar_so/ers/ers0508/EKF0508.pdf." 73-80.
- 26 Hammer, B. V., T. (2003). Mathematical Aspects of Neural Networks. The European Symposium on Artificial Neural Networks. Europa: 12.
- 27 Herrera Chica, I. (2012). Conjunto de Objetos de Aprendizaje para la Matemática Discreta. La Habana, Universidad de las Ciencias Informáticas: 140.
- 28 Horruitiner Silva, P. (2007). "INTRODUCCIÓN DEL LIBRO LA UNIVERSIDAD CUBANA: EL MODELO DE FORMACIÓN." Revista Pedagogía Universitaria. XII(4): pp.183.
- 29 J. Goldstein, P. (2005) "The uses of management information and technology in higher education." 8, 107.
- 30 Johnson, L. (2011). THE HORIZON REPORT. THE HORIZON REPORT. <http://www.educause.edu/library/resources/2011-horizon-report>, The New Media Consortium: 40.

- 31 Johnson, L. (2012). THE HORIZON REPORT. THE HORIZON REPORT. ISBN 978-0-9846601-3-1: 42.
- 32 Martínez Labaut, T. (2008). Propuesta de técnicas de aprendizaje de elementos Elementos Virtuales. La Habana, Universidad de las Ciencias Informáticas: 119.
- 33 Martínez Labaut, T. (2008). Propuesta técnica de aprendizaje de elementos virtuales. Facultad. La Habana, Universidad de las Ciencias Informáticas: 119.
- 34 Martínez Sánchez, N. (2008). Modelo para diseñar Sistemas de Enseñanza-Aprendizaje Inteligentes utilizando el Razonamiento Basado en Casos. Ciencia de la computación. Santa Clara, Universidad Central Marta Abreu. MSc: 116.
- 35 Mell, P. (2011) "The NIST Definition of Cloud Computing." 7.
- 36 Mesa Vázquez, J. (2011). Las TIC y el profesorado. Retos ante la multiformidad de mediaciones en los escenarios escolares. Revista académica semestral. <http://www.eumed.net/rev/ced/28/jmv.htm>. 3: 3.
- 37 Morejón González, L. (2010). Propuesta de un Sistema de Análisis de la Información Académica para Toma de Decisiones. La Habana, Universidad de las Ciencias Informáticas. Diploma: 73.
- 38 Portuondo, R. (2001) "Didáctica para escuelas preparatorias."
- 39 Ríos Rodríguez, L. R. (2008). Enseñando Prolog con mapas conceptuales. Enseñando Prolog con mapas conceptuales. <http://rcci.uci.cu/index.php/rcci/article/view/49>, Universidad de las Ciencias Informatáticas. 2: 27.
- 40 Ríos Rodríguez, L. R. (2009). Ambiente de enseñanza-aprendizaje inteligente para la programación lógica. Ciencias de la computación. Santa Clara, Universidad Central de las Villas Marta Abreu. MSc: 120.
- 41 Ruiz del Olmo, F. J. (2010). Conocimiento en la nube: características sociocomunicativas del cloud computing. <http://redalyc.uaemex.mx/src/inicio/ArtPdfRed.jsp?iCve=199514908030>: 17.
- 42 Soledad González, C. (2004). SISTEMAS INTELIGENTES EN LA EDUCACIÓN: UNA REVISIÓN DE LAS LÍNEAS DE INVESTIGACIÓN Y APLICACIONES ACTUALES. Revista Electrónica de Investigación y Evaluación Educativa. <http://www.uv.es/RELIEVE> 10.
- 43 Tedeschi, N. (2011) "¿Qué es un Patrón de Diseño?", 13.
- 44 Welicki, L. (2009). Patrones y Antipatrones: una Introducción. :