

Universidad de las Ciencias Informáticas

Facultad 1



Título: Análisis y diseño de propuesta de sistema automatizado para el control del proceso docente a distancia del Centro de Identificación y Seguridad Digital (CISED).

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

Autor: Yanet Velazquez Jimenez

Tutores: Ing. Anie Bermudez Peña

Ing. Eduardo Manuel Macías Sotolongo

Ciudad de La Habana, junio 2010

“Año 52 de la Revolución”



“Podrán cortar todas las flores, pero nunca marchitarán la primavera.

Pues mi confianza en el triunfo final de lo que creo, es completa”

Ernesto Che Guevara

Dedico este trabajo de diploma a toda mi familia, que ha sido siempre la guía en mi formación como persona y como profesional. Y en especial:

A mi mamá por tantas noches de desvelo, de cuidado y amor. Por ser más que una madre una amiga que siempre está a mi lado, que me brinda apoyo y me da fuerzas para seguir adelante.

A mi papá por ser el ejemplo donde me refugio, la meta a cumplir, la alegría de mis éxitos porque son pensando en él.

A mi hermana que vea en mí el ejemplo a seguir, una amiga con la que puede contar siempre.

A mis abuelas, Amparo y Neida por ser las mejores abuelas preocupadas, dedicadas y compasivas.

A Maritza que en todos estos años de convivencia se ha convertido en mi segunda mamá, una amiga con la que sé que siempre puedo contar, por estar siempre pendiente de mí y que está orgullosa de poder tenerme como una hija.

A mi abuelito que donde quiera que esté, estará orgulloso de que ya pueda ser una profesional como siempre quiso.

Agradezco principalmente a mis padres y mi hermana por ayudarme en todo momento, por confiar en mí.

A Maritza por brindarme su amor y cariño, por ser una guía durante todos estos años.

A la Revolución y a la Universidad por darme la posibilidad de ser alguien en la vida.

A Isabel por todo el apoyo y el cariño que siempre me brindó.

A mis tías y tíos, por estar pendientes de mí en cada momento, por aconsejarme y darme ánimo para seguir adelante.

A todos mis amigos que siempre me han apoyado en los buenos y malos momentos en estos cinco años de la carrera.

Aquella persona que significó mucho en estos cinco años de mi carrera, que siempre estuvo a mi lado y me estimuló a seguir adelante en los momentos más difíciles.

A Geidy, Suley, Yeny, Yamaráis, Mariela y Albert por ser más que amigos, mis hermanos y apoyarme durante estos cinco años de Universidad.

A mi tutor y amigo Eduardo, que confió en mí y me brindó su ayuda durante el desarrollo de este trabajo, además por haberme tenido tanta paciencia.

A mis asesores Aneyty, Dolenni, Yadira y Albin por el apoyo que me brindaron.

A todos.....MUCHAS GRACIAS.

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo al Centro de Identificación y Seguridad digital, y la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Yanet Velazquez Jimenez

Eduardo Manuel Macías Sotolongo

En la actualidad los avances alcanzados por el hombre en las Tecnologías de la Información y las Comunicaciones (TIC) han aumentado su nivel de vida, dotándolo de poderosas herramientas para desarrollar de manera más eficiente muchos de los procesos de la vida cotidiana. En este sentido, en la educación han tomado un extraordinario auge el uso de herramientas informáticas y entornos virtuales de aprendizaje a distancia, los cuales amplían el horizonte de posibilidades dotando tanto a profesores como a estudiantes de la posibilidad de intercambiar en espacios no convencionales o tradicionales, pero igualmente válidos para desarrollar el proceso docente de forma exitosa y gestionar de una mejor manera las competencias individuales.

La Universidad de las Ciencias Informáticas (UCI) promueve un nuevo modelo de formación basado en tres líneas fundamentales de trabajo: Docencia, Producción e Investigación. Con el objetivo de fusionar de forma armónica estos procesos se han creado varios centros productivos en la universidad, tal es el caso del Centro de Identificación y Seguridad Digital (CISED).

El objetivo principal de la presente investigación es desarrollar el análisis y diseño de una herramienta informática que permita crear un ambiente virtual de enseñanza y aprendizaje a distancia en el CISED. Para realizar el diseño de la misma se utilizó la metodología “Proceso Unificado de Desarrollo de Software” (RUP), UML (*Unified Modeling Language*) como lenguaje de modelado y Visual Paradigm como herramienta CASE. Además se proponen herramientas para la posterior implementación de la aplicación: NetBeans como entorno de desarrollo, PostgreSQL como Sistema Gestor de Base de Datos, como servidor Web Apache y PHP5 como lenguaje de programación. Para el diseño propuesto se utilizaron los patrones de diseño *grasp* y *gof*, y el estilo arquitectónico MVC (Modelo, Vista, Controlador).

Palabras clave: Aprendizaje, Competencias, Enseñanza, Entorno virtual.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	6
1.1 Introducción	6
1.2 Definiciones conceptuales del estudio.	6
1.2.1 Gestión por competencia	7
1.2.2 Acreditación y Certificación por competencia.....	7
1.3 Sistemas de enseñanza y aprendizaje a distancia.....	8
1.3.1 E-learning	8
1.3.2 Sistemas de gestión de contenidos.....	9
1.3.3 Sistemas de gestión de aprendizaje	10
1.3.4 Sistemas de gestión de contenido de aprendizaje	11
1.4 Análisis de otras soluciones existentes	11
1.4.1 Blackboard.....	11
1.4.2 ATutor.....	12
1.4.3 Claroline	12
1.4.4 Dakeos	13
1.4.5 Moodle.....	14
1.5 Fundamentación de las metodologías utilizadas	17
1.5.1 Metodologías tradicionales	18
1.5.2 Metodologías ágiles.....	19
1.6 Tecnologías usadas para la realización del software	21
¿Por qué UML?.....	24
1.6.5.1 Lenguajes del lado del cliente	25
Lenguaje de Marcas de Hipertexto (HTML).....	25
1.6.5.2 Lenguajes del lado del servidor	26
¿Por qué utilizar PHP?.....	26
1.7 Lenguaje de Marcas Extensible (XML).....	27
1.8 Servidor Web	27
1.8.1 Apache	27

¿Por qué utilizar el Servidor Apache?	28
1.9 Plataforma de desarrollo	28
1.9.1 NetBeans v6.8	28
1.10 Conclusiones	28
Capítulo 2: Características del Sistema.....	29
2.1 Introducción	29
2.2 Objeto de automatización	29
2.3 Propuesta solución	30
2.4 Modelo de dominio.....	31
2.4.1 Diagrama de clases del dominio	31
2.4.2 Definición de las clases del Modelo de Dominio	32
2.5 Especificación de los requisitos de software	32
2.5.1 Requisitos funcionales	32
2.5.2 Requisitos no funcionales	35
2.6 Modelado del Sistema	37
2.6.1 Actores del sistema.....	37
2.6.2 Definición de los casos de uso.....	37
2.6.3 Listado de Casos de Usos del sistema:	37
2.6.4 Diagrama de casos de uso del sistema.....	38
2.6.5 Especificación de los casos de uso.....	39
2.7 Estimación del esfuerzo	41
2.7.1 Cálculo de Puntos de Casos de Uso sin ajustar:.....	41
2.7.2 Cálculo de Puntos de Casos de Uso ajustados.....	43
2.8.1 Factor de complejidad técnica	43
2.8.2 Factor de ambiente	44
2.7.3 Estimación del esfuerzo por Puntos de Casos de Uso.....	45
2.7.3.1 Cálculo del esfuerzo del proyecto:.....	45
2.8 Validación del sistema propuesto.....	46
2.9 Conclusiones	51
Capítulo 3: Análisis y Diseño.....	52
3.1 Introducción	52
3.2 Arquitectura del sistema.....	52

3.3	Patrones de diseño	53
3.4	Análisis	55
3.4.1	Modelo de Clases del Análisis	55
3.4.2	Clases del análisis	56
3.4.3	Diagrama de clases del análisis	57
3.4.4	Diagrama de interacción de clases del análisis	58
3.5	Diseño	58
3.6	Diagramas de interacción del diseño	59
3.6.1	Diagramas de interacción del Diseño	59
3.7	Diagrama de clases del diseño	59
3.8	Diseño de la base de datos	62
3.8.1	Diagrama Entidad Relación de la base de datos	62
3.8.2	Diagrama de despliegue	63
3.9	Conclusiones	64
	Conclusiones	65
	Recomendaciones	66
	Bibliografía	67
	Glosario de Términos	69

Índice de figura

<i>Fig. 1</i>	<i>Diagrama de clases del modelo de dominio</i>	<i>31</i>
<i>Fig. 2</i>	<i>Diagrama de casos de uso del sistema</i>	<i>38</i>
<i>Fig. 3</i>	<i>Representación del patrón MVC</i>	<i>53</i>
<i>Fig. 4</i>	<i>Atributos esenciales y estereotipos de una clase del análisis</i>	<i>56</i>
<i>Fig. 5</i>	<i>DCA_Gestionar Certificaciones</i>	<i>57</i>
<i>Fig. 6</i>	<i>DCA_Buscar Usuario</i>	<i>57</i>
<i>Fig. 7</i>	<i>DCA_Definir nota del curso</i>	<i>57</i>
<i>Fig. 8</i>	<i>DCA_Gestionar Usuario</i>	<i>58</i>
<i>Fig. 9</i>	<i>DCA_Gestionar requisitos de vencimiento de cursos</i>	<i>58</i>
<i>Fig. 10</i>	<i>DCA_Sugerir recurso</i>	<i>58</i>
<i>Fig. 11</i>	<i>CU_Gestionar certificación</i>	<i>59</i>
<i>Fig. 12</i>	<i>CU_Buscar Usuario</i>	<i>60</i>

Fig. 13 CU_Gestionar Usuario.....	60
Fig. 14 Definir nota del curso.....	61
Fig. 15 Gestionar requisito de vencimiento de curso.....	61
Fig. 16 CU_Sugerir recurso.....	62
Fig. 17 Diagrama entidad relación de la base de datos.....	63
Fig. 18 Diagrama de despliegue.....	64

Índice de tabla

Tabla. 1 Definición de las competencias.....	6
Tabla. 2 Actores del sistema.....	37
Tabla. 3 CU_Buscar Usuario.....	39
Tabla. 4 CU_Gestionar requisitos de vencimiento de cursos.....	39
Tabla. 5 CU_Definir nota del curso.....	40
Tabla. 6 CU_Gestionar certificación.....	40
Tabla. 7 CU_Gestionar usuario.....	40
Tabla. 8 CU_Recomendar tema.....	41
Tabla. 9 CU_Sugerir recurso.....	41
Tabla. 10 Factor de Peso de los Actores sin ajustar.....	42
Tabla. 11 Factor de Peso de los Casos de Uso sin ajustar.....	42
Tabla. 12 Factor de complejidad técnica.....	44
Tabla. 13 Factor de ambiente.....	44
Tabla. 14 Esfuerzo total en horas-hombre.....	46

Introducción

El desarrollo de las Tecnologías de la Información y las Comunicaciones, desde fines del pasado siglo, ha logrado extender el uso en el mundo de dispositivos avanzados que integran múltiples funcionalidades de almacenamiento, procesamiento y transferencia de información. De tal forma que no hay una esfera del conocimiento científico, social o económico que no las aplique. La comunicación y la información representan un paradigma para el crecimiento cultural y económico de la sociedad, quedando demostrado el nivel de integración alcanzado y el cambio que gradualmente se está produciendo en muchas esferas de la misma, así como el que se avecina en la educación por el empleo de estas técnicas novedosas.

Como consecuencia de la aplicación de estas tecnologías al ámbito de la educación surge el *e-Learning*, definido como un conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, proporcionando el acceso a la información y a la comunicación con otros participantes.

Para la implementación de este tipo de estrategias formativas resultan de especial interés las plataformas de gestión de contenido/conocimiento (*Content Management Systems o CMS*), que en la actualidad se emplean comúnmente para facilitar la gestión de la publicación y distribución de contenidos y conocimientos a través de la web, por lo que también son conocidos como gestores de contenido Web (*web Content Management o WCM*).

En Cuba, el sistema educativo no pudo quedar al margen de los nuevos cambios tecnológicos, con el empleo de las TIC los centros de educación han evolucionado de forma ascendente en los últimos años, propiciando una serie de transformaciones entre las que se encuentran la creación de nuevos modelos para la formación, el uso de herramientas informáticas y entornos virtuales de aprendizaje a distancia.

La Universidad de las Ciencias Informáticas es una universidad con características únicas en el país, que dentro de sus misiones está producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación; en la misma, estudiantes, profesores y trabajadores son capaces de promover el desarrollo de la economía del país a través de los grandes proyectos de informatización. Dentro de la UCI, se hace un gran esfuerzo por fomentar la industria cubana del software, con énfasis en

la identificación y seguridad digital, la salud, la educación, las telecomunicaciones, el turismo, la cultura, etc.

El avance de la UCI en la esfera de las TIC es imparable, así como en el perfeccionamiento de los procesos educativos a través de entornos virtuales de aprendizaje. Estos eliminan la exigencia de espacios íntegramente presenciales y capaces de asegurar una continua comunicación entre estudiantes y profesores de la universidad. Un ejemplo de ello lo constituye el Entorno Virtual de Aprendizaje (EVA), que permite a los alumnos realizar diferentes actividades evaluativas y la comunicación entre usuarios a través del foro o la sala de chat, viabilizando la enseñanza de forma presencial, a distancia o combinando ambas modalidades, facilitándole al profesor los resultados obtenidos por sus estudiantes.

En el 2005, después de un estricto análisis realizado para la evaluación de diferentes plataformas de teleformación, teniendo en cuenta las características y condiciones de uso en la Universidad se comienza a utilizar el MOODLE (Entorno de Aprendizaje Dinámico Modular, Orientado a Objetos, por sus siglas en inglés de *Modular Object Oriented Dynamic Learning Environment*), una plataforma gratuita de e-learning de gran flexibilidad; técnicamente, Moodle es una aplicación que pertenece al grupo de los Gestores de Contenidos Educativos (LMS, *Learning Management System*), también conocidos como Entornos de Aprendizaje Virtuales (VLE, *Virtual Learning Managements*), un subgrupo de los Gestores de Contenidos (CMS, *Content Management System*). (Santo, 2007)

Posteriormente en el 2008 surge el Centro de Identificación y Seguridad Digital el cual fue concebido en su núcleo por los principales desarrolladores del Sistema Administrativo de Identificación Migración y Extranjería (SAIME) y fue nutrido en su totalidad por estudiantes y profesores pertenecientes a la Facultad 1 de la UCI. En el mismo se desarrollan productos que pretenden garantizar la identidad de todo ciudadano usando tecnologías de última generación en la informática, como documentos electrónicos y modernos dispositivos de control biométrico

Este Centro cuenta con mucha información diseminada y esto trae consigo que en ocasiones por desconocimiento se incurran en investigaciones repetitivas, desgaste de tecnologías, conocimientos científicos y en la mayoría de los casos en capital humano, acarreando pérdida de tiempo real para el

desarrollo de los proyectos. El Centro ha aumentado su potencial investigativo, con trabajos de gran impacto y novedad, por lo que se necesita una herramienta que facilite la retroalimentación de este conocimiento y que gestione la información académica procesada en el CISED permitiendo de esta forma acreditar y certificar los aportes de los recursos humanos. Es necesario además realizar pruebas de aceptación o captación para integrantes nuevos al Centro y mantener un dinámico ambiente para la gestión de competencias individuales.

Por la problemática antes expuesta se define como **Problema Científico** de la investigación: ¿Cómo gestionar la enseñanza a distancia y las competencias individuales en el Centro de Identificación y Seguridad Digital?, el **Objeto de estudio** se centra en los procesos de enseñanza-aprendizaje a distancia y la gestión de competencias individuales, para lo cual el **Campo de Acción** está orientado a los procesos de enseñanza-aprendizaje a distancia para estudiantes y profesores del CISED.

La investigación que se desarrolla tiene como **Objetivo general**: Diseñar una herramienta capaz de gestionar el proceso de aprendizaje a distancia y los procesos de competencias en el CISED.

Como **Objetivos específicos** se plantean los siguientes:

- Identificar los sistemas de software libre y multiplataforma existentes en el mundo que permiten la gestión de la enseñanza a distancia.
- Fundamentar el uso de metodologías, tecnologías y herramientas de desarrollo de software a utilizar.
- Definir los procesos que permitirán el desarrollo de la herramienta informática para la gestión de las competencias individuales del personal del CISED.
- Definir un diseño con las funcionalidades que deben estar presentes en la herramienta.
- Diseñar los componentes por casos de usos.

Las **tareas de la investigación** a desarrollar para cumplir los objetivos trazados son:

- Evaluar los sistemas para la enseñanza a distancia existente en el ámbito extranjero y nacional.
- Estudiar documentos sobre los Lenguajes de Modelado, Metodologías de desarrollo, Herramientas de modelado y seleccionar la adecuada para el desarrollo de la aplicación.

- Realizar el diseño teórico metodológico de la investigación.
- Precisar el lenguaje de programación a utilizar.
- Especificar la herramienta de modelado.
- Realizar modelo de dominio.
- Realizar Levantamiento de requisitos.
- Confeccionar los diagramas de clases del análisis.
- Confeccionar los diagramas de clases de diseño.
- Diseño de los prototipos no funcionales.

Para dar respuesta a la interrogante planteada en el problema científico se parte de la siguiente **Idea a defender**: Con el diseño de una herramienta que permita el control del proceso de enseñanza y la gestión por competencias individuales, se ofrece una solución para elevar la calidad en el proceso docente a distancia que realiza el CISED.

Métodos científicos de investigación:

En el desarrollo de la investigación se utilizan varios métodos científicos, como son:

Métodos teóricos:

- **Método Histórico - Lógico:** Se usa para analizar los diferentes sistemas de aprendizaje a distancia y poder seleccionar el que cumpla con los requerimientos del CISED.
- **Método Analítico - Sintético:** Mediante este método se analizaron las teorías y documentos para extraer los elementos más importantes relacionados con el objeto de estudio, en los sistemas de enseñanza y aprendizaje a distancia que existen en el mundo.
- **Método de la modelación:** Mediante este método se lleva a cabo la modelación de procesos que permiten un análisis previo de la situación, sin ser ejecutados en la realidad.

Métodos empíricos:

- **Método Entrevista:** Permite recopilar información del surgimiento y expectativas futuras del CISED.

Los **Resultados Esperados** de este trabajo son:

- El diseño de un sistema de software libre y multiplataforma para gestionar la enseñanza a distancia en el CISED.

- El diseño de una herramienta que permita la gestión de competencias individuales en CISED.

El presente documento consta de tres capítulos:

- **Capítulo 1:** Fundamentación Teórica: En este capítulo se hace un estudio de diferentes sistemas de enseñanza a distancia que nos ofrecen elementos que sirvieron de referencia para la selección del sistema de gestión de contenido para el CISED. En esta parte se expone el respaldo teórico de los temas tratados en el informe y de la solución planteada. Además se analizan diversos lenguajes y herramientas para el desarrollo e implementación de la aplicación que se propone.
- **Capítulo 2:** Características del Sistema: Consiste en la descripción de los requerimientos funcionales y no funcionales con las condiciones que debe cumplir el sistema. Se presenta un modelo de negocio que nos permite comprender la estructura y los problemas actuales del CISED. Al mismo tiempo, se definen los procesos, actores, responsabilidades y diagramas con sus respectivas descripciones de los diferentes casos de uso del sistema.
- **Capítulo 3:** Análisis y Diseño: Se muestran los diagramas de clases del análisis y el diseño para cada uno de los casos de uso del sistema y al mismo tiempo los diagramas de interacción, además del diseño de la base de datos y la descripción de cada una sus tablas. Es una visión general del sistema donde se observa el cumplimiento de los objetivos.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se abordan los fundamentos teóricos relacionados con los sistemas de enseñanza/aprendizaje a distancia. Además, se describen los principales conceptos asociados al dominio del problema, las principales aplicaciones y herramientas utilizadas para el desarrollo del trabajo.

1.2 Definiciones conceptuales del estudio.

Los cambios en el mundo actual unido al desarrollo de las ciencias han impactado con fuerza en la esfera profesional, posibilitando que el término de competencias sea de interés fundamental en las organizaciones; para elevar su nivel productivo a través de las evaluaciones de competencia en el desempeño de actividades definidas.

Las mismas son utilizadas para dar mayor idoneidad al capital humano en pos de los objetivos del área u organización; como también desarrollar el intelecto del ser humano. Por tanto, una de las formas más comunes de agilizar el proceso de enseñanza, es proponiendo las competencias entre los estudiantes. Esto conlleva a la superación entre todos los involucrados, además fomenta y estimula el conocimiento de los implicados.

David Mclelland define las competencias como: “Las características subyacentes de una persona que están causalmente relacionadas con los comportamientos y la acción exitosa en su actividad profesional”. (Higuira, 2008)



Tabla. 1 Definición de las competencias

En el contexto en que se enmarca la presente investigación se entenderá por competencias todas aquellas habilidades, conocimientos y cualidades que permiten al individuo superarse en el desempeño de diferentes roles que serán ocupados en dependencia de su nivel competitivo en las certificaciones, para de esta forma lograr mayor productividad en el desarrollo de proyectos productivos.

- **Habilidades:** Destreza y experiencia adquirida durante el aprendizaje de cada individuo (estudiante), son las actividades que el individuo puede realizar de forma destacada producto a la práctica y el conocimiento.
- **Cualidades:** Rasgos del carácter de los individuos que lo inclinan a realizar diferentes tareas o acciones para su aprendizaje de forma destacada. Las mismas son generadas por las motivaciones y conocimientos del mismo.
- **Conocimiento:** Conjunto de información o posesión de datos interrelacionados que el individuo adquirió mediante la experiencia o aprendizaje.

1.2.1 Gestión por competencia

Es un modelo de gerenciamiento que permite evaluar las competencias específicas que requiere un puesto de trabajo de la persona que lo ejecuta, además, es una herramienta que permite flexibilizar la organización, ya que logra separar la estructura del trabajo de la gestión de las personas, introduciendo a estas como actores principales en los procesos de cambio de las empresas y finalmente, contribuir a crear ventajas competitivas de la organización. (Muñoz, y otros, 2001)

1.2.2 Acreditación y Certificación por competencia

Luego de consultar abundante bibliografía relacionada con el concepto de "Acreditación", entre varios autores sobresale la Dr. Marta Pujada que define la acreditación como el "proceso por el cual un profesional o un especialista en formación, demuestra o satisface un nivel de competencia y calidad adecuado.

La certificación es el reconocimiento público y temporal de las competencias adquiridas dentro o fuera de las instituciones educativas para ejercer funciones profesionales o laborales. (G., 2008)

1.3 Sistemas de enseñanza y aprendizaje a distancia.

1.3.1 E-learning

Es un sistema de educación a distancia o literalmente significa aprendizaje electrónico, en el mismo se integra el uso de tecnologías de la información y diferentes elementos didácticos para la formación y capacitación de los usuarios o estudiantes en línea, haciendo uso de los servicios y herramientas que esta tecnología provee.

Es el aprendizaje asistido por tecnologías de la información. El *e-Learning* fomenta el uso intensivo de las TIC facilitando la creación, adopción y distribución de contenidos, así como la adaptación del ritmo de aprendizaje y la disponibilidad de las herramientas de aprendizaje independientemente de límites horarios o geográficos. Permitiendo al alumno intercambiar opiniones y aportes a través de las TIC. (Tecnoeditor, 2007)

Para los educadores, *e-Learning* es el uso de tecnologías de redes y comunicaciones para diseñar, seleccionar, administrar, entregar y extender la educación. (Comunidad educativa boliviana, 2010)

El *e-Learning* como sistema educativo o formación a distancia ofrece las siguientes ventajas:

1. Fácil creación, adopción y distribución de contenidos, así como la adaptación del ritmo de aprendizaje y la disponibilidad de las herramientas de aprendizaje independientemente de los límites de horarios o ubicación geográfica (Almira, 2010). Propiciando una constante comunicación e intercambio de conocimiento entre los usuarios, a través de las TIC.
2. La gestión del conocimiento garantiza el mejoramiento y enriquecimiento de los procesos de aprendizaje.
3. Puede contar con herramientas informáticas y diversos medios como internet/intranet, los chats, wiki, correo electrónico capaz de permitir la comunicación entre estudiantes y profesores, así como útiles de presentación de contenidos tales como CD-ROM y producciones multimedia (textos, animaciones, gráficos, videos).
4. Permite una superior efectividad en los costos debido a la reducción de los gastos en viajes y la flexibilidad de incluir estudiantes al curso cuando esto sea necesario, sin ningún gasto asociado.

Actualmente estos sistemas de aprendizaje son una de las mejores opciones para llevar a cabo el proceso docente a distancia de forma continua, constituyendo una alternativa en la formación de profesionales que combinan trabajo y modernización sin necesidad de acudir a un aula permanentemente.

Teniendo en cuenta esto se puede decir que en los sistemas *e-learning* existen dos bloques fundamentales:

- 1.WBT (*Web Based Training*) o *Courseware*: Los contenidos de un curso en *e-learning*, los materiales educativos en diferentes formatos multimedia (Muras, 2009).
- 2.LMS (*Learning Management System*) i/o CMS (*Content Management System Web Based Training*): Las plataformas o sistemas de gestión de contenidos para el aprendizaje que son las herramientas informáticas que permiten la visualización y uso de los contenidos y la gestión de los participantes, alumnos y tutores (Muras, 2009).

El progreso de los CMS durante los últimos años revela varios aspectos reseñables, tales como: la aparición de los sistemas; LMS (Sistemas de Gestión de Aprendizaje) y los LCMS (Sistemas de Gestión de Contenido de Aprendizaje) .Los cuales a pesar de tener nombres y acrónimos tan similares, pueden crear una cierta confusión, y sus funciones son muy diferentes.

1.3.2 Sistemas de gestión de contenidos

Un sistema de gestión de contenido, es un *software* que se utiliza principalmente para facilitar la gestión de Web, ya sea en Internet o en una intranet, y por esto también son conocidos como gestores de contenido Web (*web content management o WCM*) (Plataformas abiertas de e-learning para el soporte de contenidos, 2007).

Estos sistemas están creados para contener el ciclo de vida de los contenidos: creación, mantenimiento, gestión, actualización y publicación de contenido digital en diferentes formatos. El objetivo que persiguen estos programas informáticos; por una parte es la generación de la información y por otra su administración y difusión. Además, la petición del usuario e interacción con el servidor permite generar sitios Web dinámicos, con el formato predefinido y el contenido extraído de la base de datos del servidor.

1.3.3 Sistemas de gestión de aprendizaje

Es un término amplio usado para referirse a sistemas que organizan y proveen acceso a servicios de aprendizaje en línea para estudiantes, educadores y administradores. Estos servicios incluyen control de acceso, fuente de contenidos de aprendizaje, herramientas de comunicación y organización en grupos de usuarios. También se les conoce como plataformas de aprendizaje.

Todas las plataformas LMS brindan funcionalidades que permiten: gestionar usuarios, recursos y actividades de formación, administrar el acceso de usuarios, controlar y hacer seguimiento del proceso de aprendizaje de los estudiantes, realizar evaluaciones, generar informes y gestionar diversos servicios de comunicación rápida entre usuarios para el intercambio de experiencias y conocimientos.

Por lo que es necesario que un LMS posea las siguientes características básicas:

- **Perfiles de acceso:** La formación en línea permite la administración de usuarios donde cada uno de ellos presenta diferentes roles y privilegios de acceso a funcionalidades.
- **Gestión de usuarios:** Funcionalidad de suma importancia de un software ya que a través de ella podemos tener control de los usuarios, obteniendo su aprendizaje e historial, así como la generación de informes.
- **Herramientas de comunicación:** La comunicación es uno de los aspectos importantes en la formación a distancia, la cual permite intercambiar conocimientos, experiencias y habilidades entre alumno /profesores y alumno /alumno. Debido a esto es que todo e-learning debe contener diversos sistemas de comunicación tanto asíncrona¹ como síncronos².
- **Sistema de Gestión de Calificaciones:** Permite obtener los resultados obtenidos en los exámenes objetivos, así como las evaluaciones que el profesor realiza durante el curso verificando conocimientos adquiridos. Muestra calificaciones y actas de notas después de una evaluación en línea. La puntuación de actividades del curso se puede realizar individual y por grupos, incluyendo lista de espera.
- **Registro de Participación y Asistencia de los diferentes usuarios:** Registra un historial de usuario con la asistencia (servicios visitados, fechas en las que fue visitado y el tiempo de estancia en las diferentes áreas) y participación en las diferentes actividades.

¹Síncrono: interlocutores que coinciden en tiempo.

²Asíncrono: Interlocutores que no coinciden en tiempo.

- **Servicios y áreas configurables:** Cada curso, cada tipo de alumno, cada materia, puede requerir diferentes elementos o recursos. Es importante que los servicios y áreas que compongan la plataforma sean configurables dependiendo de cada curso concreto.

1.3.4 Sistemas de gestión de contenido de aprendizaje

Los Sistemas de gestión de contenido de aprendizaje son sistemas que integran las funcionalidades de un LMS y un CMS, es decir, combinan las dimensiones administrativas y de soporte del LMS con las dimensiones de crear y combinar contenido. Además los LCMS pueden o no utilizar el lenguaje estándar XML para el intercambio de datos entre aplicaciones software y básicamente Web.

El aprendizaje a distancia es un aspecto que en la actualidad ha ido evolucionando de tal forma que requiere de recursos que permitan tanto la creación de cursos como la distribución de contenidos integrados en la misma plataforma, ya que permite a los desarrolladores crear, almacenar, reutilizar, gestionar, distribuir contenidos de aprendizaje y controlar la eficacia del proceso de aprendizaje de una forma sencilla, rápida y eficiente.

1.4 Análisis de otras soluciones existentes

Actualmente existen una gran cantidad de plataformas, tanto comerciales como de código abierto. Las cuales han sido de impacto para la sociedad facilitando la educación a distancia. No existe una mejor que otras, si no que debemos seleccionar la que más se asemeje a las características que la organización requiere, es por ello que a continuación se hacen algunas comparaciones de diferentes plataformas con características LMS y LCMS.

1.4.1 Blackboard

Esta plataforma educativa se comenzó a utilizar en la Universidad de Cornell en el año 1997. Es una plataforma que se caracteriza por administrar un conjunto de recursos que permiten desarrollar cursos virtuales, específicamente: impartir y distribuir contenidos que se encuentran representados en diversos formatos (texto, sonido, video y animación), realizar evaluaciones en línea, llevar a cabo el seguimiento académico de los alumnos participantes, asignar tareas y desarrollar actividades en ambientes colaborativos (*Blackboard Academic Suite*). Ofrece herramientas para la interacción sincrónica y asincrónica, así como la publicación de contenidos. Sus características son similares a las de Moodle, a

diferencia que no es gratuito y patenta sistemáticamente todo aquello que renueva. Por lo que es desechada a la hora de realizar la selección.

Otras plataformas de software libre a tener en cuenta son aplicaciones como Claroline, Dokeos, ATutor y Moodle que a continuación se analizarán.

1.4.2 ATutor

ATutor es un LCMS de código abierto y de fácil manejo y administración por parte de alumnos y profesores. Este proyecto empezó en 2002 en la Universidad de Toronto, por lo que la accesibilidad ha sido una prioridad en su diseño y cumple todos los estándares en este sentido. (Atutor, 2010)

ATutor es un programa diseñado en PHP y en un ambiente tipo LAMP (Linux + Apache + MySQL + Php), trabaja sobre plataformas Windows, GNU/Linux, Unix, Solaris, soporte a 32 idiomas, contiene herramienta de Gerencia y administra alumnos, tutores, cursos y evaluaciones en línea, herramienta de autoría y de colaboración incorporada. Esta plataforma de formación en línea cuenta con un sistema de autenticación que le permite tener cursos abiertos a los que se puede ingresar como invitado, y cursos cerrados para los que hay que registrarse previamente. La incorporación de las especificaciones de empaquetado de contenido IMS/SCORM, permite que los diseñadores de contenidos creen contenido reutilizable que se puedan intercambiar entre diversos sistemas de aprendizaje.

ATutor posee mensajería interna, permitiendo la comunicación, pero no permite apuntes en línea, al contrario de moodle que permite crear directorios, subir archivos a los cuales los estudiantes tienen acceso, además de facilitar adjuntar archivos una vez concluido un examen a los estudiantes. La comunidad de desarrollo es grande pero no alcanza a la comunidad de desarrollo de la plataforma moodle.

1.4.3 Claroline

Es una plataforma de aprendizaje y trabajo virtual (*eLearning* y *eWorking*) de origen francés, herramienta de código abierto y software libre que permite a los formadores construir eficaces cursos en línea y gestionar las actividades de aprendizaje y colaboración en la web. La misma es capaz de albergar

fácilmente gran cúmulo de usuarios. Es un *groupware*³ asíncrono y colaborativo. Sigue las especificaciones de estándares como SCORM⁴ e IMS⁵.

La plataforma Claroline está organizada alrededor del concepto de espacios relacionados con un curso o actividad pedagógica. Cada espacio provee una lista de herramientas que permite crear contenidos de aprendizaje y gestión/manejo de actividades de formación. (Atutor, 2010)

Claroline en su última versión consta de nuevas funcionalidades las cuales se muestran a continuación:

- Está bajo la licencia GNU GPL⁶.
- Traducido a 35 idiomas.
- Creación de Ejercicios Online.
- Claroline, basado en herramientas y lenguajes libres como PHP (recomendado mayor 5.1.6), El servidor de bases de datos MySQL posterior a 4.23 y como servidor Web recomendado Apache.
- Es compatible con los entornos de Linux, Mac OS X⁷ y Windows.
- Cuenta con espacios públicos para establecer una conversación asíncrona de distintos temas y herramientas en línea para el debate (conversación sincrónica).
- Organiza los grupos de trabajo a partir de los usuarios matriculados en este curso y la definición de los escenarios de matrícula. Esta característica también la posee Moodle y Dakeos.
- Supervisar usuarios y estadísticas permite hacer el seguimiento del acceso a la plataforma de sus usuarios y las herramientas que se usan.

1.4.4 Dakeos

Es una aplicación de administración de contenidos de cursos y una herramienta de colaboración. Iniciado como una derivación de Claroline (a fines de 2003), está traducido en más de 30 idiomas y usado por más

³ Groupware: En español (conjunto de programas informáticos colaborativos). Son métodos y herramientas de software que permiten realizar trabajos colectivos entre usuarios a través de la red.

⁴SCORM: Modelo Referenciado de Objetos de Contenido Compatible (Sharable Content Object Reference Model, SCORM).

⁵ IMS: Information Management System, es un gestor de bases de datos jerárquicas y un gestor transaccional con alta capacidad de proceso

⁶GNU/GPL: La Licencia Pública General de GNU, en inglés General Public License (GNU). Es una licencia creada por la Free Software Foundation.

⁷ Mac OS X: Sistema Operativo.

de mil organizaciones. Está optimizado para trabajar con tecnología LAMP⁸ (Linux, Apache, MySQL y PHP) (Campus AulaGlobal, 2004). Especialmente recomendada a usuarios que tengan nociones mínimas de computación y con mayor interés por el contenido de los cursos.

Un aspecto a tener en cuenta es que Dakeos solo utiliza MySQL como gestor de base datos. Además, esta plataforma requiere configuraciones especiales en el servidor Web igual que Moodle, pero necesita adicionar soportes para las herramientas como los videos conferencia y el buzón de tareas.

- Las videoconferencias: Permite la comunicación a través de una webcam. Además, puede transformar archivos de PowerPoint o Impres en presentaciones visibles para los alumnos. Todas estas características dependen de la implantación de un servidor Flash Red5.
- Buzón de tareas: Básicamente facilita el intercambio de archivos entre alumnos y el docente del curso: El creador del curso puede enviar archivos a uno o a muchos estudiantes; los estudiantes pueden enviar archivos al creador del curso y también enviar archivos entre sí. Además, los archivos pueden incluir comentarios.

1.4.5 Moodle

Características generales

Es un entorno virtual de aprendizaje diseñado para facilitar la creación de cursos en línea. Moodle se distribuye gratuitamente como Software Libre bajo la Licencia Pública. Pertenece a los LMS, es una plataforma educativa la cual se basa en las ideas del constructivismo social (el conocimiento se construye en la mente del estudiante a través del aprendizaje colaborativo, en lugar del estudio a través de libros o enseñanza presencial); permitiendo que se convirtiera en una de las plataformas de aprendizaje a distancia más extendidas y usadas en el mundo, con una amplia comunidad de usuarios.

Moodle contiene varias herramientas de comunicación que dan la opción de configurar un curso con Foros (debates en la Web), Chat (charla en tiempo real), Glosario (vocabulario creado en común), documentos compartidos, calendarios, wiki (construcción de una Web en común), encuestas, cuestionarios, archivos en formatos variables y tareas o recursos añadidos al curso, Taller (cada alumno es evaluado por todos los demás).

⁸ LAMP: Conjunto de subsistemas de software necesarios para alcanzar una solución global.

Su numerosa comunidad de usuarios es lo que determina que tenga una vasta base de datos de problemas y soluciones para sus distintas versiones disponibles en sus foros oficiales, wiki y sistema de registro de incidentes. Los archivos pueden subirse y manejarse en el servidor, o pueden ser creados sobre la marcha usando formularios Web (de texto o HTML). (Moodle Docs, 2006)

Los paquetes de idiomas permiten la localización completa de cualquier idioma y pueden ser editados usando un editor integrado. Dado el enfoque pedagógico de Moodle, está diseñado para brindar todos los recursos y que el usuario decida cuáles de ellos utilizar.

Estas plataformas *e-learning* permiten que cada estudiante tenga su ritmo de trabajo, estableciendo plazos de entrega de las actividades las cuales son monitoreadas y evaluadas permanentemente por el profesor. Ayuda a los educadores a crear comunidades de aprendizaje en línea.

Un profesor que opera desde este punto de vista crea un ambiente centrado en el estudiante que le ayuda a construir ese conocimiento con base en sus habilidades y conocimientos propios en lugar de simplemente publicar y transmitir la información que se considera que los estudiantes deben conocer. (Elizabeth, 2008)

Bases Pedagógicas de Moodle

Moodle está cimentado en un modelo pedagógico de constructivismo social, basado en 4 conceptos principales y subyacentes:

- **Constructivismo.** Este punto de vista mantiene que la gente construye activamente nuevos conocimientos a medida que interactúa con su entorno. Todo lo que usted lee, ve, oye, siente y toca se contrasta con su conocimiento anterior y si encaja dentro del mundo que hay en su mente, puede formar el nuevo conocimiento que se llevará consigo. Este conocimiento se refuerza si puede usarlo con éxito en el entorno que le rodea. (MoodleDocs, 2007)
- **Construccionismo.** El construccionismo explica que el aprendizaje es particularmente efectivo cuando se construye algo que debe llegar a otros. Esto puede ir desde una frase hablada o enviar

un mensaje en internet, a artefactos más complejos como una pintura, una casa o un paquete de software. (MoodleDocs, 2007)

- **Constructivismo Social.** Esto extiende las ideas anteriores a la construcción de cosas de un grupo social para otro, creando colaborativamente una pequeña cultura de artefactos compartidos con significados compartidos. Cuando alguien está inmerso en una cultura como ésta, está aprendiendo continuamente acerca de cómo formar parte de esa cultura en muchos niveles. (MoodleDocs, 2007)
- **Conectados y Separados.** Esta idea explora más profundamente las motivaciones de los individuos en una discusión. Un comportamiento separado es cuando alguien intenta permanecer 'objetivo', se remite a los hechos y tiende a defender sus propias ideas usando la lógica buscando agujeros en los razonamientos de sus oponentes. El comportamiento constructivo es cuando una persona es sensible a ambas aproximaciones y es capaz de escoger una entre ambas como la apropiada para cada situación particular. (MoodleDocs, 2007)

Moodle posee como ventajas fundamentales:

- **Alta Disponibilidad:** es suficientemente robusto como para satisfacer las diversas necesidades de miles de estudiantes, administradores, creadores de contenidos y profesores simultáneamente. (Moodle Docs, 2006)
- **Escalabilidad:** es posible ampliar su infraestructura o escalar para resolver el futuro crecimiento, tanto en términos de volumen de contenidos educativos como del número de estudiantes. (Moodle Docs, 2006)
- **Facilidad de uso:** Apoya un conjunto de servicios automatizados y personalizados, tales como aprender a ritmo individual y perspectivas específicas de aprendizaje, el acceso, la entrega y la presentación de materiales son fáciles de utilizar y muy intuitivos como navegar por la Web. (Moodle Docs, 2006)
- **Interoperabilidad:** es capaz de intercambiar información utilizando estándares abiertos de la industria para implementaciones WEB como: (Moodle Docs, 2006)
 - Autenticación por LDAP, Búsquedas directas en bases de datos externas, basada en protocolo Shibboleth, o también utilizando IMAP, NNTP, CAS o FirstClass.

- Moodle admite la utilización de un servidor LDAP (por ejemplo el Directorio Activo de Microsoft), y el estándar IMS *Enterprise* (a través de un *plugin* que se puede descargar) para su matriculación en cursos.
- Admite la importación/exportación de Objetos Reutilizables de Aprendizaje empaquetados de acuerdo con los estándares IMS *Content Packaging* y SCORM.
- Las preguntas de los cuestionarios pueden ser exportadas en el formato estándar internacional IMS QTI 2.
- En Moodle, los canales de noticias RSS pueden integrarse en un sitio WEB completo o un curso.
- Se puede acceder a las discusiones de los foros como noticias RSS, y por lo tanto integrarse en otros sistemas o sitios WEB con funcionalidad RSS.

¿Por qué Moodle?

La plataforma seleccionada como modelo de formación para el CISED es Moodle, en primer lugar porque las herramientas del sistema garantizan la seguridad y protección de los datos, ya que los usuarios tienen acceso sólo a los permisos del rol asignado, además, cuenta con una amplia y activa comunidad universitaria de desarrollo y mantenimiento en todo el país; ejemplo de ellas es la UCI, donde varios proyectos han desarrollado módulos que son compatibles con Moodle, de acuerdo con las necesidades y características del modelo de formación de la universidad; en segundo lugar por ser un entorno que apoya un conjunto de servicios automatizados y personalizados centrados en la comunicación y las actividades de enseñanza-aprendizaje, así como lo suficientemente robusta como para satisfacer las diversas necesidades de los usuarios, brinda herramientas para la gestión de contenido educativo que facilitan el aprendizaje del individuo a partir de la secuenciación de actividades en línea y el auto estudio.

1.5 Fundamentación de las metodologías utilizadas

Para guiar el proceso de desarrollo de un software es imprescindible el uso de una metodología, la cual será capaz de guiar al equipo de trabajo con el objetivo de aumentar la calidad y la eficacia en el proceso de generación de software. Entre las principales metodologías que guían el desarrollo de un software se

encuentran las metodologías ágiles (XP⁹ y Scrum¹⁰) y las tradicionales (RUP¹¹ y MSF¹²). De ellas sólo analizamos una de cada metodología.

1.5.1 Metodologías tradicionales

Las metodologías tradicionales se centran principalmente en el control del proceso de desarrollo del software, estableciendo rigurosamente la definición de roles, actividades, artefactos que debe producirse, herramientas y notaciones que se usarán para el modelado y la documentación detallada, con el fin de obtener un software eficiente y predecible. No son adaptables a cambios, por lo que no son métodos convenientes cuando se trabaja en un entorno, donde los requerimientos no pueden predecirse o pueden estar propensos a modificaciones. Esta metodología hace énfasis en una detallada planificación del trabajo total que se realizará y hasta que no sea completada no se puede comenzar el ciclo de desarrollo del producto software.

Dentro de esta clasificación encontramos a RUP (Proceso Unificado de Desarrollo). Es actualmente la metodología más utilizada y conocida mundialmente, la estudiada en la Universidad de las Ciencias Informáticas.

1.5.1.1 Proceso Unificado de Desarrollo (RUP)

RUP es uno de los procesos de desarrollo de software más generales de los existentes en la actualidad, garantiza la elaboración de todas las fases en el desarrollo de un producto orientado a objetos y pensado para adaptarse a las características de cualquier proyecto. Entre las características principales de RUP podemos encontrar:

- **Iterativo e incremental:** Es la secuencia de iteraciones que pueden realizarse en forma de cascada. Cada iteración aborda una parte de la funcionalidad, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura.
- **Dirigido por casos de uso:** No son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los mismos constituyen un elemento

⁹ XP: Extreme Programming

¹⁰ Scrum: Metodología ágil para la gestión y desarrollo de software.

¹¹ RUP: Rational Unified Procces

¹²MTF: Microsoft Solution Framework

integrador y una guía del trabajo que comprende todos los flujos de trabajo: requerimientos, análisis, diseño, implementación y prueba.

- **Centrado en la arquitectura:** La arquitectura se representa mediante varias vistas que se centran determinados aspectos del sistema. Está formado por las vistas lógicas, de implementación, de proceso y de despliegue, más la de Casos de Uso que es la que da cohesión a todas las vistas y juntas forman el llamado modelo 4+1 de la arquitectura.

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales. Los primeros seis son conocidos como flujos de ingeniería que abarcan: los flujos de trabajo Modelo de Negocio, Requerimientos, Análisis, Diseño, Implementación y Prueba; los últimos tres son de soporte: Administración del Proyecto, Administración de Configuración y Cambios y Ambiente.

El desarrollo de un software realizado con la metodología RUP se divide en 4 fases:

- **Inicio:** Se determina la visión del proyecto y se describe el negocio del mismo.
- **Elaboración:** Determina la óptima arquitectura del sistema y se obtiene una aplicación ejecutable, la cual responde a los requerimientos funcionales.
- **Construcción:** Obtenemos el producto listo para su utilización, con al menos una versión del software pasado por pruebas. Además de estar documentado y tiene manual de usuario.
- **Transición:** La versión del software ya está listo para su instalación. Esto puede implicar reparación de errores.

1.5.2 Metodologías ágiles

La filosofía de las metodologías ágiles enmarca su trabajo en el producto software y en el factor humano, proporcionándole mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas, pues define que es mejor formar un buen grupo de trabajo, y que éste configure su propio entorno de desarrollo en base a sus necesidades. Son efectivas en proyectos donde los requerimientos están en constantes cambios y cuando se exige reducir drásticamente los tiempos de desarrollo, conservando una alta calidad en el producto final. Entre las metodologías ágiles más difundidas en la actualidad se encuentra XP (*Programación Extrema*).

1.5.2.1 Programación Extrema (XP)

XP es una metodología de desarrollo ligera. Está pensada para obtener un software eficiente en un reducido tiempo de trabajo después de estar en constantes cambios los requisitos del software. A continuación se muestran algunas características que diferencian a XP de las demás metodologías:

- Está concebida para ser usada dentro de proyectos pequeños y de corto plazo.
- La disponibilidad de personal es poca. Se basa en equipos de desarrollos pequeños; (2 a 10) personas trabajando en un mismo sitio y con pocos roles, facilitando intercambiar responsabilidades y trabajar en conjunto.
- Consiste en una programación rápida o extrema que está basa en la sencillez, la comunicación y la realimentación o reutilización del código desarrollado.
- Puesto que puede ser un proyecto para uso interno, aunque también puede utilizarse para otros proyectos.
- La arquitectura del proyecto se va definiendo y mejorando a lo largo de su desarrollo.
- El cliente forma parte del equipo de desarrollo .Con la utilización de XP la comunicación con el cliente es a través de diálogos cara a cara, el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo y lograr un producto que satisfaga sus necesidades.
- Uno de los principios básicos de esta metodología es el cambio frecuente de los requerimientos, y su fácil adaptación a los proyectos.
- Las pruebas de comprobación se escriben en fases tempranas del desarrollo, antes de comenzar la implementación permitiendo dirigir el proceso de desarrollo software.

¿Por qué RUP?

Una vez analizadas las diferentes metodologías, se selecciona RUP para desarrollo del presente trabajo, debido a su capacidad de aumentar la productividad en equipo y proporcionar una amplia experiencia en el proceso de desarrollo de software, a todos sus miembros. Permite que la producción del producto cumpla con las necesidades de los usuarios a través de la especificación de los requisitos, proporcionando extensas guías y ejemplos para las actividades críticas.

1.6 Tecnologías usadas para la realización del software

1.6.1 Rational Rose

Rational Rose es una herramienta que integra todos los elementos que propone la metodología RUP para cubrir el ciclo de vida de un proyecto (Inicio, elaboración, construcción y transición). Construida por los creadores de UML (Booch, Rumbaugh y Jacobson) permitiéndole a los desarrolladores y arquitectos del software representar el sistema a través de un lenguaje común, para comprender y comunicar la estructura y la funcionalidad del mismo en su fase de construcción, es el uso de la notación estándar en la arquitectura de software UML para expresar gráficamente el modelado de los esquemas de un sistema software.

Cada integrante del equipo de proyecto puede modelar sus componentes e interfaces de forma individual y luego unirlos con otros componentes del proyecto, gracias a que cada cual tiene sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue). Es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Permite el diseño detallado del software, generación de código fuente de programas y bases de datos, así como la ingeniería inversa donde podemos obtener el diseño del software a partir de su código fuente.

1.6.2 Visual Paradigm

Visual Paradigm es una herramienta que utiliza UML como lenguaje de modelado, el cual facilita la rápida construcción de aplicaciones de calidad y a un menor costo. Logra la integración con Eclipse/IBM WebSphere, Borland Jbuilder y NetBeans IDE/Sun ONE, entre otros.

Esta herramienta está diseñada para desarrollar software con programación orientada a objeto, cubre el ciclo de vida completo del desarrollo brindando ayuda a arquitectos, analistas, diseñadores y desarrolladores. Permite representar diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Visual Paradigm ofrece: (Download, 2007)

- Soporte de UML versión 2.1.

- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Interoperabilidad con modelos UML2.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas EJB - Visualización de sistemas EJB.
- Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones.
- Diagramas de flujo de datos.
- Generación de bases de datos. Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas. Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Editor de figuras.

¿Por qué Visual Paradigm?

Se decidió emplear como herramienta para realizar la Ingeniería de Software a Visual Paradigm, pues la Universidad ha incrementado los niveles de aceptación por su robustez, usabilidad y portabilidad. Además, se cuenta con una licencia, resultando factible el desarrollo desde el punto de vista económico. Su integración con el entorno de desarrollo NetBeans provee al desarrollador de un ambiente completamente funcional, permitiendo ahorrar tiempo durante el proceso de desarrollo del software, principalmente durante la fase de construcción. Soporta múltiples usuarios trabajando en un mismo

proyecto, además de facilitar la organización, visualización, integración, diseño y despliegue de los sistemas a través de la representación gráfica.

1.6.3 Auxure RP

Es un software para diagramar e idóneo para crear prototipos y especificaciones precisas para páginas web. El mismo cuenta con características que lo hacen muy útil para la realización de prototipos de forma eficiente, permitiendo de esta forma comprender la página Web visualmente, añadiendo, eliminando y modificando los elementos fácilmente. Las páginas son organizadas en forma de árbol, permitiendo exportarlas en formato HTML y cada elemento del prototipo puede ser comentado y definido de forma exhaustiva con diferentes campos, permitiendo visualizarlos sobre la propia presentación. Este sistema facilita considerablemente la presentación de los prototipos al cliente con los requisitos capturados al inicio del proyecto. Además también permite exportar a Word todos los comentarios hechos sobre cada prototipo lo cual ayuda mucho a la hora de generar un documento explicativo.

1.6.4 Lenguaje Unificado de Modelado (UML)

UML es el lenguaje de modelado de sistemas de software más usado en la actualidad, y un estándar internacional aprobado por la OMG (Objeto de administración de grupos). Cuando se define UML no se puede hablar de métodos o guías, pues simplemente es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema intensivo de software. Este lengua ofrece una forma estándar para escribir un plano del sistema, incluyendo cosas conceptuales tales como procesos de negocios y funciones del sistema, como así también cosas concretas tales como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. (Gloria Marcela Gaviria, 2009)

UML es el primer método en publicar un meta-modelo en su propia notación, incluyendo la notación para la mayoría de la información de requisitos, análisis y diseño. Se trata pues de un meta-modelo auto-referencial (cualquier lenguaje de modelado de propósito general debería ser capaz de modelarse a sí mismo). (Gloria Marcela Gaviria, 2009)

¿Por qué UML?

Se seleccionó UML como lenguaje de modelado debido a las relevantes características que lo identifican:

- Estandarización del lenguaje.
- Visualiza, especifica, construye y documenta los artefactos de un sistema que involucra una gran cantidad de software.
- Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y preside de reglas para combinar tales elementos los elementos.
- Permite la modelación de sistemas con tecnología orientada a objetos.

1.6.5 Sistemas Gestores de BD

MySQL

El sistema gestor de base de datos MySQL es un *software* de código abierto, licenciado bajo la GPL de la GNU, muy rápido, fiable, fácil de usar y robusto. Utiliza como lenguaje de programación SQL (Lenguaje estructurado de consulta), de forma generalizada en las bases de datos relacionales. El modelo relacional permite establecer relaciones entre los datos que se encuentran contenidos en tablas.

Este sistema es mayormente indicado para aplicaciones que requieren de numerosas lecturas y pocas escrituras. El mismo carece de soporte para transacciones *rollback's* y subconsultas por lo que no es recomendable su uso con grandes bases de datos a las que se accede continuamente, pues no implementa una buena escalabilidad.

PostgreSQL

Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) de código abierto. Comenzó como un proyecto en la Universidad Berkeley de California y actualmente ofrece control de concurrencia multi-versión, soportando casi toda la sintaxis SQL incluyendo subconsultas, transacciones y funciones definidas por el usuario. Proporciona un gran número de características que sólo se encontraban en las bases de datos comerciales tales como DB2¹³ y Oracle¹⁴.

Ventajas fundamentales de PostgreSQL:

¹³DB2: Marca comercial, propiedad de IBM, bajo la cual se comercializa un sistema de gestión de base de datos.

¹⁴Oracle: Herramienta cliente/servidor para la gestión de base de datos.

1. **Extensible:** El código fuente se encuentra disponible para todos sin costo alguno. Si se necesitara extender o personalizar, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales.
2. **Multiplataforma:** Está disponible para cualquier sistema Unix¹⁵ y una versión nativa de Windows que actualmente se encuentra estado beta de pruebas.
3. **Diseñado para ambientes de alta concurrencia:** PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC (Multiversión de control de Concurrencia) para conseguir una respuesta más eficiente en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones. (Pérez., 2008)
4. **Alta concurrencia:** Mientras un proceso escribe en una tabla, otros accedan a la misma sin necesidad de bloqueos.
5. **Integridad de los datos:** claves primarias, llaves foráneas con capacidad de actualizar en cascada o restringir la acción y restricción no vacía. (Pérez., 2008)
6. **Resistencia a fallas:** Escritura de registros anticipada para evitar pérdidas de datos en caso de fallos por: Energía, Sistema Operativo, Hardware. (Pérez., 2008)

¿Por qué utilizar PostgreSQL?

Se seleccionó PostgreSQL como sistema gestor de base de datos pues es capaz de manejar rutinas y reglas complejas como es la declaración de consultas SQL, control de concurrencia multi-versión, soporte a multiusuario, transacciones, optimización de consultas, herencia y arreglos. Soporta operadores definidos por el usuario y utiliza la tecnología MVCC para evitar los bloqueos innecesarios en el sistema. Presenta una excelente escalabilidad y rendimiento bajo grandes cargas de trabajo, así como una mayor protección en la integridad de los datos.

1.6.6 Lenguajes de programación

1.6.5.1 Lenguajes del lado del cliente

Lenguaje de Marcas de Hipertexto (HTML)

HTML es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos

¹⁵ Unix: Sistema Operativo, registrado oficialmente como UNIX®.

tales como imágenes. Puede describir la apariencia de un documento e incluir un *script* (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de HTML. Es compatible con navegadores reconocidos como *Internet Explorer*, *Opera*, *Firefox*, *Netscape* y *Google Chrome*.

1.6.5.2 Lenguajes del lado del servidor

Procesador de Hipertexto (PHP)

PHP es un lenguaje interpretado de alto nivel que se ejecuta en un servidor Web y permite la generación de páginas dinámicas. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos y plataformas sin costo alguno. Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas Web dinámicas:

Características fundamentales del lenguaje PHP:

- Soporte de conexiones a bases de datos tales como: *MySQL*, *PostgreSQL*, *Oracle*, *Informix*, entre otras. (Hinostroza, 2007)
- Interactúa con uno de los servidores de Web más potentes como el Apache.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de *Acrobat Reader*) hasta analizar código XML. (Hinostroza, 2007)
- Utiliza las sesiones de HTTP, conectividad de Java, LDAP, SNMP e IMAP.
- Tiene manejo de excepciones (desde PHP5).
- No se restringe a una metodología para programar, el programador puede utilizar cualquier técnica de desarrollo que le permita escribir el código de forma ordenada, estructurado y adaptable.

Las principales características de PHP son: rapidez; facilidad de aprendizaje; soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos. (Hinostroza, 2007)

¿Por qué utilizar PHP?

Se concluye que PHP será el lenguaje de programación a utilizar para desarrollar la solución propuesta, debido a las ventajas que suponen las características descritas anteriormente y al ser Moodle desarrollado en dicho lenguaje.

1.7 Lenguaje de Marcas Extensible (XML)

Es un metalenguaje extensible que permite la creación de etiquetas propias, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y organizaciones.

Al usar XML en el desarrollo de aplicaciones Web se derivan un gran número de beneficios entre los que se pueden mencionar: un mejor contexto de búsqueda, flexibilidad en aplicaciones Web y un formato estándar para la entrega de datos en la Web. Además, permite la integración de datos no compatibles entre sí, sin un formato predefinido, debido a que XML es auto descriptible. Una vez que los datos XML son recibidos, se pueden separar para transformarlos y presentarlos de la manera que se desee, garantizando que las aplicaciones sean mucho más poderosas, así como ver los mismos datos en múltiples ocasiones sin tener que acudir al servidor tantas veces para obtenerlos. Otra de las ventajas que proporciona es la actualización de datos sin necesidad de retransmitirlos en su totalidad; sino sólo aquellos que han de cambiarse.

1.8 Servidor Web

1.8.1 Apache

Es un servidor Web flexible, rápido y eficiente, continuamente actualizado y adaptado a las últimas versiones de protocolos. Es capaz de funcionar en la mayoría de las plataformas y entornos existente. Está entre los servidores Web más reconocidos a nivel internacional, debido a la capacidad de correr en múltiples Sistemas Operativos. Apache es una tecnología gratuita de código abierto y altamente configurable, de diseño modular capaz de interpretar diversos lenguajes de programación entre los que se encuentran Perl y PHP. También, permite personalizar la respuesta ante los posibles errores que pueden surgir en el servidor.

El servidor Apache está estructurado en módulos, los cuales pueden clasificarse en tres categorías específicas:

- **Módulos Base:** Son funciones básicas del Apache. (schermbEEK, 2008)
- **Módulos Multiproceso:** Responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender las peticiones. (schermbEEK, 2008)
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor se puede añadir sin necesidad de volver a instalar el *software*.

¿Por qué utilizar el Servidor Apache?

Se decide seleccionar Apache como servidor Web para la aplicación que se desarrollará, por poseer grandes ventajas que hacen que millones de servidores reiteren su confianza, las excelentes prestaciones y las características descritas anteriormente. Además de interpretar PHP, lenguaje en el cual está implementado Moodle y por ser un servidor robusto, de alta estabilidad y configurabilidad en la creación y gestión de logs.

1.9 Plataforma de desarrollo

1.9.1 NetBeans v6.8

Se propone NetBeans como herramienta para el desarrollo del sistema, pues contiene las condiciones necesarias para que los desarrolladores de software puedan crear aplicaciones Web y móviles, a través de numerosos lenguajes de programación. Es un producto gratis y de código abierto que ofrece a los desarrolladores numerosas ventajas en la creación de nuevas aplicaciones multiplataforma. Ofrece características de soporte PHP ampliado que expande el soporte de los lenguajes dinámicos con apoyo para PHP 5.3.

Esta plataforma permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos, los cuales dan la posibilidad de ser extendidos agregándole otros. Su flexibilidad entre plataformas, el cumplimiento de UML y la capacidad de administrar la complejidad, ayudan a garantizar que las aplicaciones cumplan con los requerimientos específicos del negocio.

1.10 Conclusiones

En este capítulo se hace una breve descripción de los diferentes sistemas de *software* libre para la enseñanza-aprendizaje a distancia, obteniendo como propuesta de solución una plataforma con las características que requiere el CISED. Se fundamenta la selección del lenguaje, las metodologías y herramientas de soporte para la creación de la aplicación Web que dará solución a los objetivos planteados. Además, se realizó el estudio de los procesos de acreditación y certificación por competencias.

Capítulo 2: Características del Sistema

2.1 Introducción

Para elaborar un *software* que permita las competencias individuales en el CISED, es necesario identificar aspectos que obtengan una solución óptima y factible.

En el presente capítulo se analizarán las necesidades del sistema informático EVACISED (Entorno Virtual de Aprendizaje para el Centro de Identificación y Seguridad Digital), especificándolas a través de los requerimientos funcionales y no funcionales. Por lo difícil que resulta identificar los procesos del negocio se realizará la modelación de dominio, identificando los conceptos fundamentales y sus relaciones. También se estudiarán detalladamente los casos de uso arquitectónicamente significativos que muestran las principales características del sistema, así como los actores que intervienen y las descripciones textuales de cada uno de ellos.

2.2 Objeto de automatización

En el Centro de Identificación y Seguridad Digital la gestión del proceso docente no se encuentra informatizada, provocando que sea complejo en ocasiones la comunicación estudiantes-profesor o estudiante-estudiante, así como la gestión por competencias individuales.

Esta situación no cumple efectivamente con la integración estudio-trabajo que este centro productivo requiere, pues precisa de un personal capacitado para el despliegue de *software* con alta calidad y capaces de realizar investigaciones novedosas de las últimas tecnologías a utilizar en la elaboración de *software*. Por tanto, se requiere fortalecer las competencias que son la muestra del conocimiento de cada persona para desempeñar una acción o actividad traducida en eficiencia y mayor desarrollo productivo en el proyecto.

Para darle solución a esta situación se decide ejecutar el proceso de aprendizaje a distancia a través del sistema EVACISED. El diseño de este módulo permitirá informatizar los procesos que darán solución a la situación problémica existente, así como el control del nivel que posee el personal del Centro.

2.3 Propuesta solución

Se propone el diseño del sistema EVACISED, el cual permite el control del proceso de gestión de enseñanza a distancia y las competencias individuales en el Centro. Este sistema consiste en una aplicación Web, diseñada como un bloque de Moodle con el propósito de añadirle nuevas funcionalidades que serán integradas y administradas por esta plataforma.

El contenido de estudio que será impartido en CISED se gestionará a través de la plataforma Moodle, sistema que contiene todos los componentes para la correcta gestión de publicación de contenidos. Estos cursos permiten al personal la superación en temas para elevar el desarrollo productivo con los últimos adelantos tecnológicos de nuestra era.

El sistema permite la certificación para los diferentes roles que se desempeñan en los proyectos productivos. La certificación de los roles alcanzados por los estudiantes está compuesta por varios cursos, los que deben vencer para acreditarse la misma, con un determinado nivel (A, B, C), el cual se obtiene en dependencia de la calificación obtenida al vencer una certificación. El acceso a los cursos de una certificación estará regido por un conjunto de restricciones definidas por el profesor en la creación de los mismo, lo que permite viabilizar las dependencias de un curso respecto a otro, debido a esto un estudiante no podrá matricularse en otro curso de la misma certificación sin antes haber vencido todos los pasos definidos como necesarios o precondiciones. Facilitando de esta forma el aprendizaje de los estudiantes cuando decida inclinarse por alguna certificación.

Los profesores son responsables de configurar la secuencia de pasos que debe tener los temas de un curso ya que se habilitarán una vez que se realice la tarea final de cada tema en el curso. De la misma manera el sistema debe ser capaz de recomendarle al estudiante diferentes cursos o certificaciones que pudieran ser de su interés. El sistema debe permitirle al profesor sugerir recursos a los alumnos que vencen el curso con la máxima calificación para que logren ampliar sus conocimientos, además que los usuarios puedan realizar una búsqueda con los datos de la persona. Algunos de esos datos son asignados al usuario manualmente por el administrador una vez que ingresen nuevos estudiantes en el CISED, debido a que son datos específicos de cada estudiante.

2.4 Modelo de dominio

El sistema a informatizar, requiere de la realización de modelo de dominio y su representación a través de conceptos relacionados entre sí, debido a que no se puede realizar Modelo de Negocio pues resulta complejo identificar claramente los procesos de negocio.

El Modelo de Dominio o Modelo Conceptual, captura los tipos de objetos más importantes que existen en el entorno donde estará el sistema. Por tanto, en nuestro modelo se especifica las relaciones existentes entre principales conceptos que se gestionan alrededor del sistema EVACISED.

2.4.1 Diagrama de clases del dominio

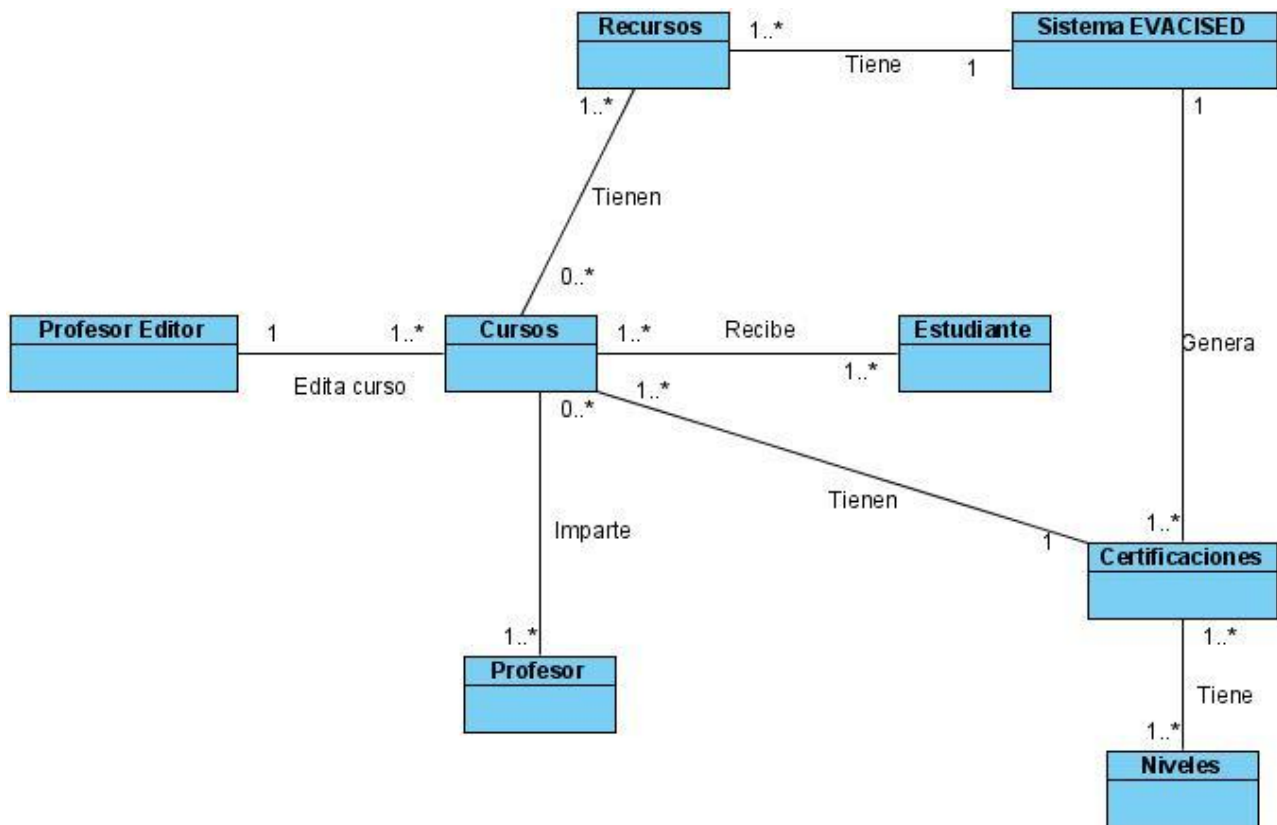


Fig. 1 Diagrama de clases del modelo de dominio

2.4.2 Definición de las clases del Modelo de Dominio

Sistema EVASISD: Entorno virtual de aprendizaje para el centro de Identificación y Seguridad Digital, sistema informatizado que será integrado a Moodle como un bloque del mismo, permitiendo de esta forma controlar el proceso docente educativo que se efectúa en el CISED.

Estudiante: Persona que podrá interactuar con el sistema y pasar determinados cursos de acuerdo con su rol. Con el vencimiento de los cursos podrá obtener niveles de certificación.

Profesor: Persona encargada de impartir y evaluar el contenido de los cursos a través de las actividades en líneas que se realicen.

Profesor editor: Persona que gestiona la creación de cursos a ser publicados, por lo que mantiene los temas y sus actividades actualizada. Además, configura el flujo de pasos que debe cumplir el estudiante para vencer el curso.

Certificaciones: Es un contenedor organizativo de cursos para los alumnos. Mediante el cual se le facilita al estudiante el trabajo en los mismos.

Niveles: Clasificación de una certificación (A, B, C) una vez certificado un determinado rol que el estudiante del Centro obtuvo a través de las competencias individuales.

Cursos: Parte fundamental de Moodle donde se muestra el contenido que el estudiante va a recibir.

Recursos: Es la información almacenada en diferentes formatos acordes a la temática de cada curso.

2.5 Especificación de los requisitos de software

En la ingeniería, los requisitos son una necesidad documentada sobre el contenido, especifica las funcionalidades del *software* que los desarrolladores deben de construir, facilitando el mecanismo apropiado para comprender lo que el cliente quiere.

2.5.1 Requisitos funcionales

Los requisitos funcionales son las acciones que los usuarios realizarán y el sistema debe ser capaz de responder a ellas. Siendo así la IEEE-STD-729 (*Institute of Electrical and Electronics Engineers*) los define como la condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Así como la condición o capacidad que debe ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. (Agudelo, 2007)

A continuación se muestran los requerimientos funcionales que debe cumplir el sistema:

RF1 Realizar búsqueda de usuarios.

RF1.1 Buscar un profesor.

- Nombre.
- Apellidos.
- Área del centro.
- Cursos impartidos.
- Provincia.
- Municipio.
- Especialidad.
- Años experiencia en su profesión.

RF1.2 Buscar un estudiante.

- Nombre
- Apellidos.
- Cursos recibidos.
- Área del centro.
- Provincia.
- Municipio.
- Certificación.
- Nivel.

RF1.3 Mostrar datos de los usuarios:

RF1.3.a Mostrar datos de los profesores.

- Nombre y Apellidos.
- Fecha de nacimiento.
- Sexo.
- Carné de Identidad.
- Apartamento UCI.
- Teléfono UCI.
- Provincia.
- Municipio.
- Especialidad.
- Área del centro.
- Años de experiencia en su profesión.

- Asignaturas que imparte.
- Cursos impartidos.

RF1.3.b Mostrar datos de los estudiantes.

- Nombre y Apellidos.
- Fecha de nacimiento.
- Sexo.
- Carné de Identidad.
- Provincia.
- Municipio.
- Apartamento UCI.
- Teléfono UCI.
- Certificaciones.
- Nivel.
- Área del centro.
- Cursos recibidos.

RF2 Realizar especificación de requisitos a cumplir para vencer un curso.

RF 2.2 Determinar temas obligatorios a cumplir para aprobar el curso.

RF 2.2 Definir las notas mínimas a obtener por temas para aprobar un curso.

RF 2.2 Configurar el sistema de cálculo de la nota del curso a partir de las notas obtenidas en cada tema.

RF2.2 Definir secuencia de temas.

RF2.2 Determinar resultado final del curso.

- Nota automática del sistema.
- Nota del profesor.

RF3 Asignar certificación al estudiante de acuerdo con el nivel alcanzado en el rol de proyecto.

RF3.1 Definir niveles de certificación.

- Nivel A.
- Nivel B.
- Nivel C.

RF3.2 Certificar usuarios que vencieron todos los cursos correspondientes al rol que se le certificará.

RF4 Sugerir recursos a estudiantes aventajados en un curso.

RF4.1 Determinar estudiantes con notas sobresalientes en un curso específico.

RF4.2 Enviar recursos para ampliar sus conocimientos.

- Ejercicios
- Bibliografía

RF5 Recomendar al estudiante otros cursos que puedan ser de su interés.

RF5.1 Estudiantes que se encuentren cursando un determinado cursos del mismo tema del que se ha creado.

RF6 Adicionar nuevos estudiantes y profesores en el sistema EVACISED.

RF7 Modificar usuarios del sistema.

2.5.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades con las que debe contar el sistema o el producto. Están vinculados a requerimientos funcionales y una vez que se conozca lo que el sistema debe hacer; se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, por lo que en muchos casos son esenciales en el éxito del producto.

A continuación se muestran los Requerimientos No Funcionales del sistema:

Requerimientos de apariencia o interfaz externa

- El sistema mostrará el nombre del producto.
- Diseño sencillo y con una interfaz amigable para el usuario, permitiéndoles la fácil manipulación sin necesidad de adiestramiento.
- Interfaz compatible con los diferentes navegadores webs existentes.
- El sistema debe poderse ejecutar en máquinas con una resolución de 800x600 y 32 bits de color y superiores. La resolución mínima recomendada es de 1024x768 y 32 bits de color.

Requerimientos de usabilidad

- El sistema será lo suficientemente sencillo como para ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web.

Requerimientos de rendimiento

- El sistema debe ser eficiente y con una velocidad de procesamiento, tiempo de respuesta y recuperación óptimo, así como un buen aprovechamiento de los recursos del servidor.

Requerimientos de Software

- Sistema Operativo: Windows o Linux.
- Servidor: Servidor Web Apache con soporte para PHP 5.
- Cliente: Navegadores como el *Internet Explorer*, *Firefox* o *Netscape*.

Requerimientos de Hardware

- Servidor: Requerimientos mínimos: 512 Mb RAM.
- Cliente: Requerimientos mínimos: 128 Mb RAM.

Requerimientos de portabilidad

- El sistema podrá ser usado bajo cualquier sistema operativo ya sea *Linux* o *Windows*.

Requerimientos de seguridad y privacidad

- Las funcionalidades del sistema serán mostradas de acuerdo con el rol que desempeña el usuario en el sistema, aprovechando el sistema de seguridad basado en roles de Moodle el cual impide al usuario manipular información que no es manejada por él. Además, los módulos y los bloques pueden crear sus propias tablas en la base de datos, por lo que no hay riesgo de que la información se pierda.
- La plataforma Moodle autentica a cada usuario para que sea identificado antes de realizar cualquier acción sobre el sistema. Si no está registrado lo forzará hasta que el mismo se inscriba en el sistema, con el propósito de proteger toda la información sensible a la visibilidad de determinados usuarios.

Requerimientos de diseño e implementación

- La herramienta para el desarrollo de la aplicación será *Netbeans* 6.8.
- Con el lenguaje de programación PHP5.
- Desarrollar sobre el LMS *Moodle*.

Capítulo 2: Características del Sistema

- La arquitectura del sistema se realizó mediante el patrón arquitectónico MVC (Modelo, Vista, Controlador).

2.6 Modelado del Sistema

2.6.1 Actores del sistema

Actores	Justificación
Profesor	Es la persona encargada de impartir los cursos y sugerirles a sus estudiantes Bibliografía, Test, etc. para su preparación ya que obtuvo las máximas calificaciones en el curso. De esta forma, se garantiza su superación.
Profesor Editor	Es la persona encargada de atender el contenido que será publicado en los cursos para la preparación de los estudiantes, así como las certificaciones que se gestionen en el centro. Además, puede participar en los fórum de debate.
Administrador	Es la persona encargada de la gestión del sistema. Tiene el control total del sistema y administra las distintas cuentas de usuarios y los roles en el sistema.
Usuario	El usuario representa a los actores del sistema, asumiendo el rol de cada uno de ellos cuando interactúan con el sistema, por tanto, funciona como un tipo de actor.

Tabla. 2 Actores del sistema

2.6.2 Definición de los casos de uso

Los casos de uso son las funcionalidades del sistema, es la forma de darle solución a la situación problémica en un lenguaje entendible por la máquina, proporcionándole al usuario un resultado significativo.

2.6.3 Listado de Casos de Usos del sistema:

CU# 1 Buscar persona: Permite buscar un usuario a través del filtrado por campos y mostrar los datos generales del mismo.

CU# 2 Gestionar Usuario: Permite al administrador del sistema crear y modificar el personal del centro que es usuario en el sistema.

CU# 3 Gestionar Requisitos de Vencimiento de Curso: Posibilita configurar y modificar los requisitos que se deben cumplir para vencer un tema. Además, después de realizarse una serie de actividades evaluativas en los temas y que determinará si el estudiante pudo o no vencer el curso. Exponiendo el estado final del estudiante en el curso.

CU# 4 Gestionar certificación: El estudiante a medida que va pasando cursos obtiene niveles de certificación(A, B, C), que les serán certificados.

CU# 5 Definir nota final del curso: El alumno se inscribe en un curso el cual puede no vencerlo. Al finalizar el mismo el profesor determinará la nota final del estudiante en dicho curso.

CU# 6 Sugerir recurso: El profesor debe ser capaz de sugerirles recursos (ejercicios, bibliografía, etc.) a sus estudiantes que presentan problemas académicos, posibilitando así superar la dificultad.

CU# 7 Recomendar tema: Exhortar al estudiante al estudio de diferentes temas que se encuentren dentro de sus temas de interés. Cuando entre a su sesión en Moodle.

2.6.4 Diagrama de casos de uso del sistema

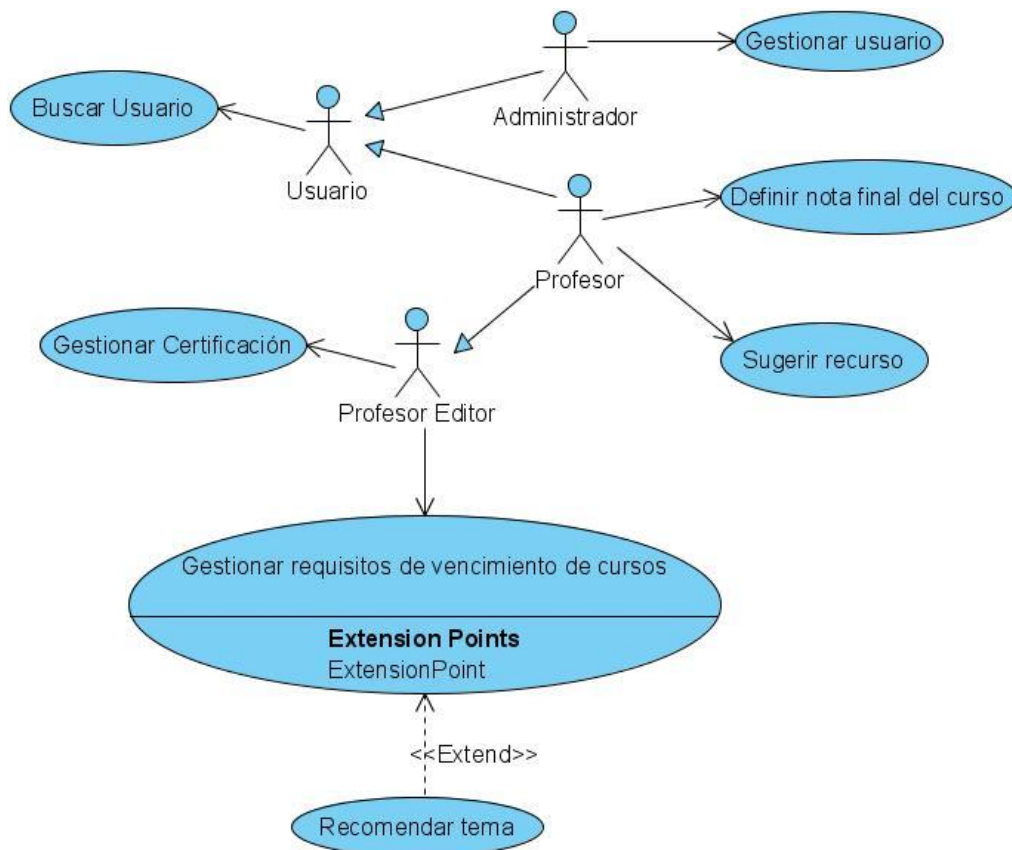


Fig. 2 Diagrama de casos de uso del sistema

2.6.5 Especificación de los casos de uso

Las descripciones extendidas ofrecen los detalles de la secuencia de acciones que se ejecutan desde el estado inicial hasta final, describiendo lo que el sistema debe hacer separando sus responsabilidades y las de sus actores.

2.6.5.1 Descripciones abreviadas de los casos de uso del sistema

Caso de Uso:	Buscar Usuario
Actores:	Usuario
Resumen:	El Caso de Uso inicia cuando el actor selecciona directamente en la interfaz principal la opción: "Buscar usuario" o una búsqueda avanzada donde el sistema muestra la interfaz correspondiente para realizar la búsqueda según los datos introducidos, que dependen del tipo de usuario que se seleccione (estudiante o profesor). Se muestran los datos del mismo, finalizando el caso de uso con la información de la persona.
Precondiciones:	El usuario debe estar registrado en el sistema
Referencias	RF-1
Prioridad	Crítico

Tabla. 3 CU_Buscar Usuario

Caso de Uso:	Gestionar requisitos de vencimiento de cursos
Actores:	Profesor Editor
Resumen:	El caso de uso inicia cuando el profesor editor selecciona la opción "Agregar un nuevo curso" y el sistema muestra la interfaz para configurar y modificar las actividades, los temas y en general el curso con requisitos que el alumno debe vencer para pasarlo. Una vez configurados los requisitos visualizaremos dicho curso permitiendo que el estudiante pueda vencer satisfactoriamente el curso. Finaliza el caso de uso.
Precondiciones:	El profesor editor debe estar autenticado en el sistema con su rol que le permite la edición del mismo.
Referencias	RF- 2
Prioridad	Crítico

Tabla. 4 CU_Gestionar requisitos de vencimiento de cursos

Capítulo 2: Características del Sistema

Caso de Uso:	Definir nota del curso
Actores:	Profesor
Resumen:	El caso de uso inicia cuando el profesor selecciona la opción: "Establecer Nota Final". El profesor puede optar por la nota final que el sistema calcula o podrá establecer la misma en dependencia de las notas obtenidas en las actividades durante el transcurso del curso y su criterio.
Precondiciones:	El profesor debe estar registrado con el rol asignado para esa actividad.
Referencias	RF-2
Prioridad	Secundario

Tabla. 5 CU_Definir nota del curso

Caso de Uso:	Gestionar Certificación
Actores:	Profesor Editor
Resumen:	El caso de uso inicia cuando el profesor editor selecciona la opción: "Gestionar certificación". A partir de ahí se le muestra las opciones de Adicionar, Modificar y Eliminar. Posteriormente introduce los datos necesarios para realizar las diferentes operaciones Crear, Modificar, Eliminar, Buscar. Finaliza el caso de uso cuando se realicen todas las operaciones deseadas y el profesor editor selecciona la opción salir.
Precondiciones:	El profesor editor debe estar autenticado en el sistema con su rol.
Referencias	RF-
Prioridad	Crítico

Tabla. 6 CU_Gestionar certificación

Caso de Uso:	Gestionar Usuario
Actores:	Administrador
Resumen:	El caso de uso inicia cuando ingresa un nuevo usuario al CISED y el administrador selecciona la opción "Crear usuario". Utiliza la autenticación que brinda LDAP e introduce algunos con datos específicos del CISED.
Precondiciones:	Debe de haberse autenticado un usuario nuevo en el centro y el administrador acceder al sistema con su rol para realizar las acciones pertinentes.
Referencias	RF-6, RF-7
Prioridad	Crítico

Tabla. 7 CU_Gestionar usuario

Capítulo 2: Características del Sistema

Caso de Uso:	Recomendar Tema
Actores:	
Resumen:	El caso de uso inicia cuando el sistema detecta que se ha creado un nuevo curso o algún usuario se inscriba en un curso y ambos usuarios están en un mismo curso.
Precondiciones:	Hay que ser usuario en el sistema con el rol de estudiante y que se halla creado el curso.
Referencias	RF-5
Prioridad	Secundario

Tabla. 8 CU_Recomendar tema

Caso de Uso:	Sugerir recurso
Actores:	Profesor
Resumen:	El caso de uso inicia cuando el actor selecciona la opción:” Sugerir recursos” y el sistema muestra la interfaz correspondiente para sugerirle una variedad de recursos que le permitan al estudiante superarse en el tema, finaliza el caso de uso.
Precondiciones:	El profesor debe estar autenticado con el rol asignado.
Referencias	RF-4
Prioridad	Secundario

Tabla. 9 CU_Sugerir recurso

2.7 Estimación del esfuerzo

Pues una vez que se determinaron los requisitos funcionales y los casos de uso con sus respectivas descripciones, se seleccionó como método de estimación “Punto de caso de uso”, el cual se define como un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. A continuación se muestran los pasos a seguir para la realización de éste método:

2.7.1 Cálculo de Puntos de Casos de Uso sin ajustar:

Los casos de uso sin ajustar se calculan a partir de la siguiente ecuación:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

Capítulo 2: Características del Sistema

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

2.7.1.1 Factor de Peso de los Actores sin ajustar

Este valor de peso de los actores sin ajustar se calcula mediante el análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Factor de Peso	Actores	UAW= $\sum(\text{actores}*\text{Peso})$
Simple	1	0	0
Medio	2	0	0
Complejo	3	4	12
Total			12

Tabla. 10 Factor de Peso de los Actores sin ajustar

2.7.1.2 Factor de Peso de los Casos de Uso sin ajustar

El factor de peso de los casos de uso sin ajustar se calcula mediante el análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. Los criterios se muestran en la siguiente tabla:

Tipo de Caso de Uso	Factor de Peso	Casos de uso	UUCW= $\sum(\text{CU}*\text{Peso})$
Simple	5	4	20
Medio	10	3	30
Complejo	15	0	0
Total			50

Tabla. 11 Factor de Peso de los Casos de Uso sin ajustar

Para finalizar se calcula los Puntos de Casos de Uso sin ajustar mediante la fórmula:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

$$\text{UUCP} = 12 + 50$$

$$\text{UUCP} = 62$$

2.7.2 Cálculo de Puntos de Casos de Uso ajustados

Después de obtenidos los Puntos de Casos de Uso sin ajustar, se debe ajustar este valor mediante la siguiente ecuación:

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{EF}$$

donde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

2.8.1 Factor de complejidad técnica

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Factor	Descripción	Peso	Valor Asignado	Total
T1	Sistema distribuido	2	4	8
T2	Objetivos de performance o tiempo de respuesta	1	3	3
T3	Eficiencia del usuario final	1	4	4
T4	Procesamiento interno complejo	1	3	3
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	5	2.5
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	0	0
T12	Provee acceso directo a terceras partes	1	3	3
T13	Se requieren facilidades especiales de	1	3	3

Capítulo 2: Características del Sistema

	entrenamiento a usuarios			
Total				53

Tabla. 12 Factor de complejidad técnica

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$\text{TCF} = 0.6 + 0.01 * \Sigma(\text{peso} * \text{valor asignado})$$

$$\text{TCF} = 0.6 + 0.01 * 53$$

$$\text{TCF} = 1.13$$

2.8.2 Factor de ambiente

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores.

Factor	Descripción	Peso	Valor	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	4.5
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en orientación a objetos	1	3	3
E4	Capacidad del analista líder	0.5	4	2.0
E5	Motivación	1	4	4
E6	Estabilidad de los requerimientos	2	4	8
E7	Personal part-time	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	3	-3
Total				17

Tabla. 13 Factor de ambiente

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$\text{EF} = 1.4 - 0.03 * \Sigma (\text{Peso}_i \times \text{Valor}_i)$$

$$\text{EF} = 1.4 - 0.03 * 17$$

$$\text{EF} = 0.89$$

Los Puntos de Casos de Uso ajustados se calculan mediante la siguiente ecuación:

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{EF}$$

$$\text{UPC} = 62 * 1.07 * 0.89$$

UPC= 59.0426

2.7.3 Estimación del esfuerzo por Puntos de Casos de Uso

donde:

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

Para obtener el factor de conversión (CF), se utiliza la fórmula que establece: tener en cuenta la cantidad de valores que afectan al factor ambiente desde (E1...E6) y están por debajo de la media (3), así como los que están por encima de la media para los restantes (E7, E8).

- Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso.
- Si el total es 3 ó 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso.
- Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Por tanto, se puede decir que: **Total_{EF} = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)**

CF=20 Horas-Hombre / Punto de Casos de uso.

E = UCP *CF

E= 66.58* 20

E= 1331,6 Horas-hombre

2.7.3.1 Cálculo del esfuerzo del proyecto:

En la siguiente tabla se muestra el esfuerzo total en Horas-hombre.

Actividad	Porcentaje %	Horas -hombre
Análisis	10	332,9
Diseño	20	665,8
Programación	40	1331,6
Pruebas	15	449,35
Sobrecarga (otras)	15	449,35

Capítulo 2: Características del Sistema

actividades)		
Total	100	3329

Tabla. 14 Esfuerzo total en horas-hombre

El presente trabajo se realiza hasta la fase de pruebas, contando con $E_T = 3329$ Horas-hombre. Estimando que el mes tiene **28 días** laborables y cada uno de ellos se trabaja **8 horas**.

Se calcula a través de la siguiente fórmula: $E_T / DL / HT = \text{mes-hombre}$

DL: Días laborales.

HT: horas de trabajo.

3339 horas-hombres / 28 días / 8 horas = 14.9 meses.

El desarrollo de las cinco etapas del proyecto tendrá una duración estimada de un año y tres meses. Mientras que las dos primeras etapas (análisis y diseño), que son las que se desarrollan en este trabajo, tendrán una duración estimada de cinco meses.

2.8 Validación del sistema propuesto

Para realizar la validación de la propuesta de prototipos no funcionales para el sistema EVACISED se realizaron listas de chequeo, las cuales fueron validadas por el equipo de calidad del proyecto. A continuación se muestran los resultados obtenidos:

La suma de rangos se obtiene a partir de los valores ya definidos para cada parámetro, se denota por P_j y se representa por:

$$P_j = \sum_{i=1}^m \frac{P_{ij}C_{ij}}{\text{cant}}$$

Donde:

P_j = Promedio a evaluar por aspectos.

P_{ij} = Peso evaluativo.

C_{ij} = Cantidad evaluativa de aspectos "j" con calificación "i".

cant = Cantidad de prototipos que fueron evaluados.

m = Máxima puntuación.

Capítulo 2: Características del Sistema

El valor de P_j se emplea para establecer la importancia entre los diferentes parámetros, de modo que el mayor valor significará una mayor importancia.

Aspectos a evaluar	Excelente(5)	Bien(4)	Regular(3)	Mal(2)	Muy mal(1)	$P_j = \sum_{i=1}^m \frac{P_{ij}C_{ij}}{cant}$
¿Es simple el vocabulario utilizado?	20	-	-	-	-	5
¿Se entienden la interfaz y su contenido?	20	-	-	-	-	5
¿Está diseñada la interfaz para facilitar la realización eficiente de las tareas de la mejor forma posible?	20	-	-	-	-	5
¿El prototipo de interfaz cubre todo lo descrito en el caso de uso?	20	-	-	-	-	5
¿La interfaz corresponde con las características, metas y nivel de experiencia de los usuarios de esta aplicación?	17	1	2	-	-	4.75
¿Está la información libre de errores gramaticales, deletreado, ortográficos y de los errores tipográficos? (estas clases de errores no solamente indican una carencia del control de calidad, pero pueden producir realmente inexactitudes en la información.)	15	5	-	-	-	4.75
En cajas de diálogo "tabuladas", los nombres de	20	-	-	-	-	5

Capítulo 2: Características del Sistema

la lengüeta o pestañas no son abreviaturas						
Los nombres de los botones de comandos no son abreviaturas.	13	2	5	-	-	4.4
¿Los campos se encuentran alineados (en forma de fila y columnas)?	20	-	-	-	-	5
Los nombres de las cajas de diálogos deben estar a la izquierda de los mismos.	20	-	-	-	-	5
Los botones deben nombrarse de manera que resulte fácil de entender las acciones para lo que están diseñados y con una descripción clara de sus funciones.	20	-	-	-	-	5
Los términos deben presentarse capitalizados (primera letra en mayúsculas)	20	-	-	-	-	5
No se deben colocar textos extensos	16	2	2	-	-	4.7
¿El prototipo de interfaz cubre todo lo descrito en el caso de uso?	20	-	-	-	-	5

Tabla. 15 Lista de chequeos de interfaz de usuario

Capítulo 2: Características del Sistema

Aspectos a evaluar	Si	No
¿Están incluidas todas las funciones requeridas por el cliente?	x	
¿Se especifican las condiciones de software y hardware necesarias para la definición del sistema?	x	
¿Se especifican las políticas de seguridad a seguir?	x	
¿Se definen las características que debe tener el sistema para lograr el mayor rendimiento, teniendo en cuenta cada una de las posibles situaciones que se puedan presentar?	x	
¿Se establecen las características que debe cumplir el sistema para brindar una mayor confiabilidad al usuario?	x	
¿Se definen las características que debe tener la interfaz del sistema?	x	
¿Se especifican las características que debe cumplir el sistema para ser fácil de utilizar por el cliente/usuario? (requisitos de usabilidad)	x	
¿Se establecen las propiedades que debe tener el sistema para que sea compatible con varios	x	

Capítulo 2: Características del Sistema

sistemas operativos? (requisitos de portabilidad)		
¿Se declaran las características que debe poseer el sistema para que esté disponible a cada uno de los usuarios que lo necesiten? (requisitos de disponibilidad)		x
¿Se exponen las características que debe tener el sistema para permitir la integridad de la información almacenada? (requisitos de integridad)		x
¿Se definen las características que debe tener el sistema para lograr su correcto funcionamiento? (requisitos de funcionalidad)	x	
¿Se establecen las restricciones del diseño a tener en cuenta?	x	
¿Los requisitos están expresados en un lenguaje apropiado para el cliente/usuario?	x	
¿Cada requisito admite tan solo una única interpretación?	x	
¿Cada característica del producto final está descrita utilizando una terminología única?	x	
¿Todos los requisitos están declarados en forma de acción	x	

Capítulo 2: Características del Sistema

(deben terminar en ar, er, ir)?		
¿Existen requisitos incompatibles con otros requisitos?		x
¿Existe facilidad para hacer cambios en los requisitos?	x	
¿El cambio o modificación de un requisito implica la modificación de otro/s relacionado/s con él?		x

Tabla. 16 Lista de chequeos de requisitos de software

Después de procesadas las evaluaciones a través del equipo de calidad del CISED se determinó que el proceso de análisis y diseño de la herramienta EVACISED propuesto cumple satisfactoriamente con las pautas analizadas.

Las rubricas cuantitativas representadas en la tabla.15 promediaron 4.9 de una escala máxima de 5, siendo la menor evaluación obtenida 4.4, lo cual expresa los resultados positivos alcanzados.

Respecto a los resultados expresados en la tabla.16 se pudo apreciar que se desarrollaron y documentaron la mayoría de los aspectos evaluados.

2.9 Conclusiones

En este capítulo se describen las características del sistema a desarrollar. Se representó gráficamente el modelo de negocio, se listaron los requerimientos que debe cumplir el software para un buen funcionamiento, así como para darle cumplimiento a la problemática existente en el CISED mencionadas en el capítulo anterior. Además, se definieron los actores y se describieron detalladamente cada uno de los casos de usos a modelar en el sistema. Finalmente se realizó la estimación del esfuerzo y tiempo de desarrollo del proyecto, así como la validación del sistema propuesto mediante listas de chequeo.

Capítulo 3: Análisis y Diseño

3.1 Introducción

El presente capítulo aborda sobre el tercer flujo de trabajo que propone la metodología RUP para el desarrollo de software, en el flujo de análisis y diseño se hace una breve descripción del sistema, modelado a través de los artefactos correspondientes. En la fase de Análisis y Diseño se especifican los diagramas de clases que describen gráficamente las especificaciones de las clases del software y las interfaces de la aplicación. Además, se modelarán los diagramas de interacción correspondientes a cada caso de uso más significativo del sistema representados con estereotipos Web, en el diseño de la base de datos se exhibe el diagrama de entidad relación y la descripción de sus tablas.

3.2 Arquitectura del sistema

Para que un sistema sea desarrollado con calidad es necesario establecer desde el inicio una arquitectura que describa los principios fundamentales arquitectónicos del sistema lo que satisface las peticiones del cliente y garantizar robustez, seguridad y escalabilidad en el sistema.

Se utilizará para el diseño arquitectónico del sistema EVACISED el patrón MVC (Modelo, Vista, Controlador). Este patrón es ampliamente utilizado en aplicaciones Web, pues garantiza la organización del código de la aplicación, separando los datos en tres componentes distintos: la interfaz de usuario (vista), la lógica de control (controlador) y el acceso a dato (modelo). Cualquier cambio producido en el Modelo se refleja automáticamente en cada una de las Vistas al mismo tiempo debido a que la conexión entre el Modelo y sus Vistas es dinámica; produciéndose en tiempo de ejecución, no así en tiempo de compilación

1. La capa de Modelo es el componente encargado del acceso a datos. Define la lógica de negocio, todo lo que tiene que ver con la persistencia de los datos. Guarda y recupera la información del medio persistente que se utilice (base de datos).
2. La Vista define la interfaz de usuario, HTML+CSS enviados al navegador. Es con lo que el usuario interactúa y presenta la información obtenida por el modelo del medio persistente.
3. El Controlador responde a eventos y modifica la vista y el modelo. Gestiona las entradas del usuario a través de una representación visual con código HTML que genera la interacción con

el Modelo por una petición que el usuario realizó. El modelo obtiene la información y la pasa a la vista devolviéndole un resultado obtenido del medio persistente.

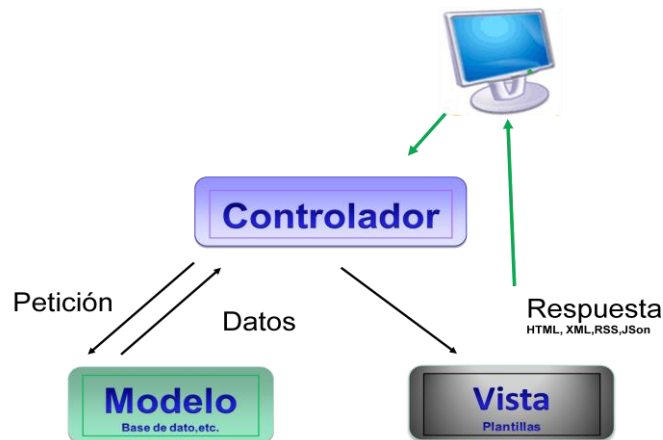


Fig. 3 Representación del patrón MVC

En aplicaciones Web, la vista es una página HTML y el código que provee de datos dinámicos a la página, frecuentemente usa el patrón MVC a pesar de ser similar a la Arquitectura en capas. A continuación se muestra las diferencias entre estos patrones en cuanto a la forma de ejecución de los componentes y su conexión:

- La Arquitectura en capas es lineal y pueden estar físicamente distribuidas, donde la comunicación entre los componentes, se establece escalonadamente por lo que la capa de presentación no se comunica directamente con la de acceso a dato, ya que tiene que pasar por una capa intermedia.
- El patrón MVC es ejecutado de forma triangular y todos sus componentes se comunican entre sí. Este patrón es importante ya que se puede dividir cada componente en otras capas en dependencia de la necesidad del usuario.

3.3 Patrones de diseño

Un patrón de diseño es una solución estándar para un problema común de programación. En un software contribuye a la reutilización de su diseño, identificando aspectos clave de la estructura de este, que puede ser aplicado en una gran cantidad de situaciones. Esta característica demuestra su gran importancia ya que reduce los esfuerzos de desarrollo y mantenimiento, mejora la seguridad, eficiencia y consistencia de los diseños, y proporciona un considerable ahorro en la inversión.

Para el desarrollo del sistema se tuvo en cuenta los patrones de asignación de responsabilidades, conocidos como patrones GRASP acrónimo de “*General Responsibility Assignment Software Patterns*”, los cuales tienen como objetivo fundamental orientar al diseñador en cómo asignar las responsabilidades a cada clase en diferentes circunstancias. (Universidad de las Ciencias Informáticas, 2010)

Además se tuvo en cuenta los patrones Gof acrónimo de “Gang of Four”, los cuales se clasifican en dependencia del propósito para los que hayan sido definidos.

Patrones GRASP

Experto: Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. El objetivo de este patrón es asignar una responsabilidad a la clase que tiene la información necesaria para cumplir dicha responsabilidad. Expresa de manera intuitiva que los objetos hacen cosas relacionadas con la información que poseen. (Universidad de las Ciencias Informáticas, 2010)

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, actividad muy frecuente en los sistemas orientados a objetos. El propósito fundamental es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Este patrón favorece el bajo acoplamiento, es decir el bajo nivel de dependencia entre las clases; el grado en que una clase puede trabajar sin recurrir a otras es menor. Por tanto, permite un aumento de la reutilización de código y minimiza el riesgo de tener que realizar muchas modificaciones en las clases cuando se deba cambiar una. (Universidad de las Ciencias Informáticas, 2010)

Bajo Acoplamiento: En las clases la herencia no debe estar muy extendida ya que la idea es que existan la menor cantidad de clases atadas entre sí. Para de esta forma poder realizar modificaciones en alguna de ellas sin que repercutan en las demás, potencializando la reutilización y eliminando las dependencias entre clases.

Alta Cohesión: Este patrón tiene como responsabilidad de que todos los elementos del diseño una única tarea dentro del sistema, no desempeñada por el resto de los elementos.

Controlador: Es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. El mismo sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

Patrón Gof

Singleton: Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. Ya que al sistema estarán conectado numerosos usuarios a la vez haciendo uso de los servicios y herramientas para la enseñanza- aprendizaje en línea.

Proxy: Proporciona un sustituto o representante de otro objeto para controlar el acceso a éste.

Iterator. Proporciona un modo de acceder secuencialmente a los elementos de un objeto agregado sin exponer su representación interna.

3.4 Análisis

Durante el modelado del análisis, se analizan con profundidad los requisitos que se describen en la captura de requisitos, refinándolos y estructurándolos. Con el objetivo de comprender detalladamente los requisitos y obtener una descripción más precisa de los mismos, que nos ayude a estructurar el sistema incluyendo su arquitectura.

3.4.1 Modelo de Clases del Análisis

El Modelo de Análisis es una aproximación al modelo de diseño, el mismo incluye las clases y sus objetos organizados en paquetes que colaboran. Entiéndase por clases de análisis a una abstracción de una o varias clases que son estereotipadas en dependencia de su funcionalidad. Los estereotipos son las clases interfaz, entidad y control, las cuales describen las realizaciones de los casos de uso.

El modelo de análisis ayuda refinar los requisitos y permite razonar sobre los aspectos internos del sistema, incluidos sus recursos compartidos internos. Proporciona una estructura no sólo útil para el mantenimiento de los requisitos, sino también que se utiliza como entrada en las actividades de diseño e implementación y se describe con el lenguaje que el desarrollador comprende.

3.4.2 Clases del análisis

Es un tipo de diagrama estático que permite describir la estructura de un software a través de clases, atributos y las relaciones entre ellos. Son utilizados durante el proceso de análisis y diseño de un sistema software.

Las clases del análisis se identifican por tres prototipos básicos:

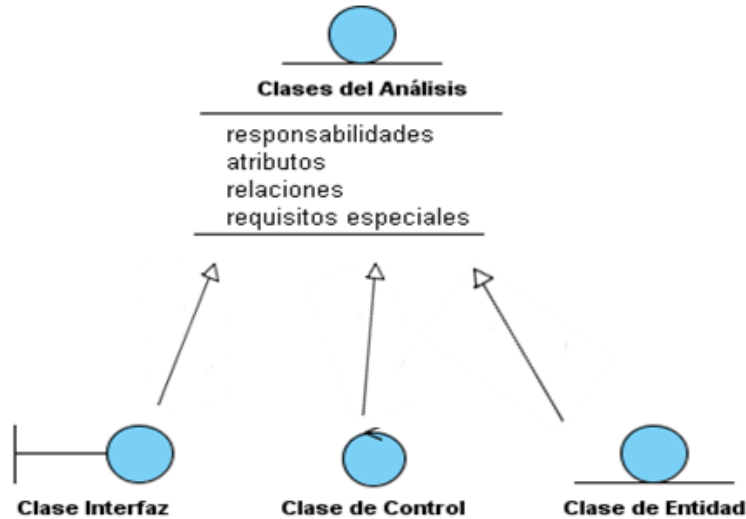


Fig. 4 Atributos esenciales y estereotipos de una clase del análisis

Interfaz: Modelan la interacción entre el sistema y sus actores.

Control: Representan coordinación, secuencia, transacciones, y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. Representan derivaciones y cálculos complejos, como la lógica del negocio, que no pueden asociarse con ninguna información concreta, de larga duración, almacenada por el sistema.

Entidad: Modelan la información que posee una larga vida y que es a menudo persistente. Además, modela el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto del mundo real, o un suceso del mundo real.

3.4.3 Diagrama de clases del análisis

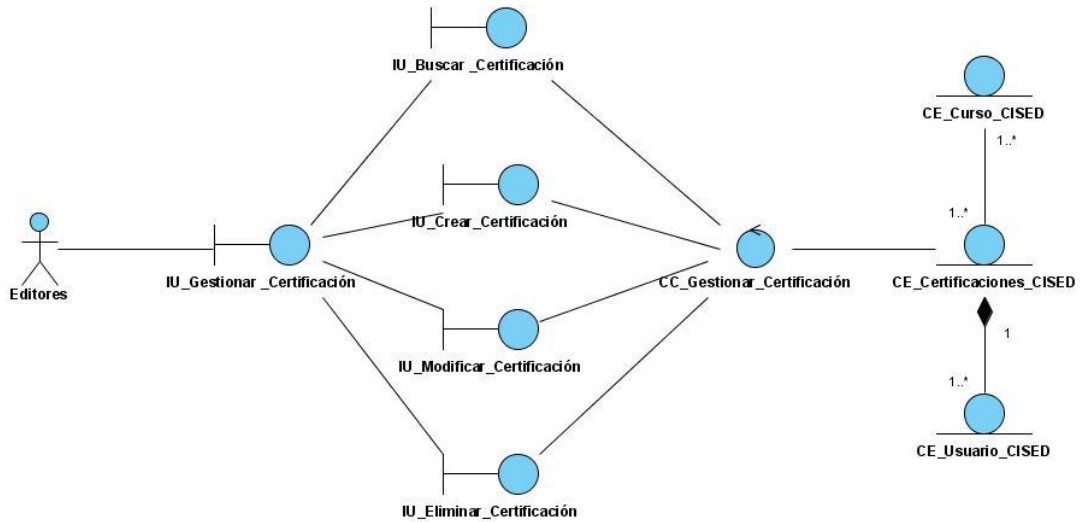


Fig. 5 DCA_Gestionar Certificaciones

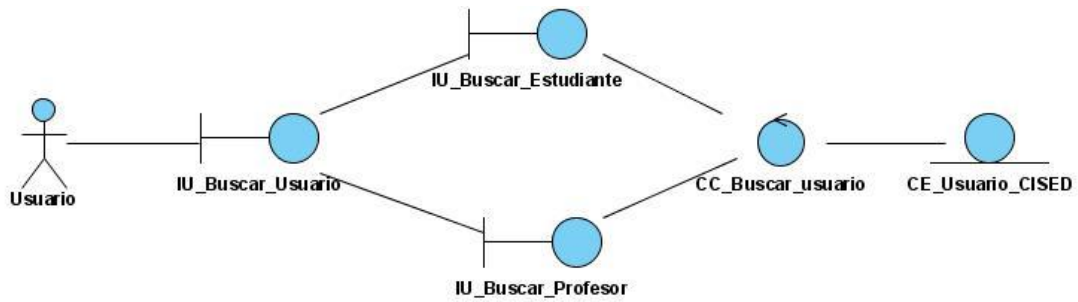


Fig. 6 DCA_Buscar Usuario

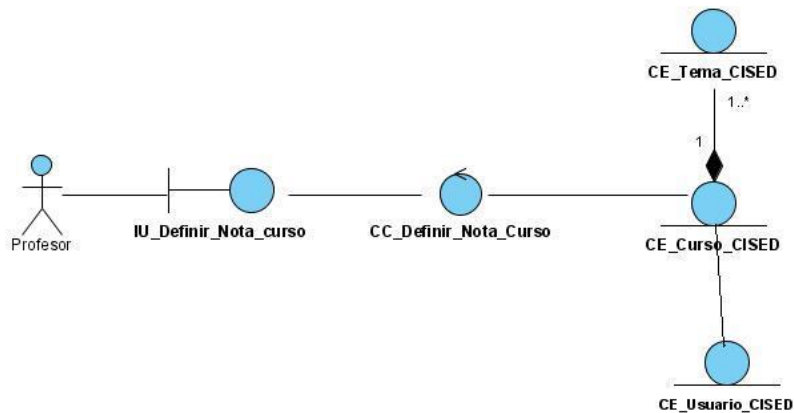


Fig. 7 DCA_Definir nota del curso

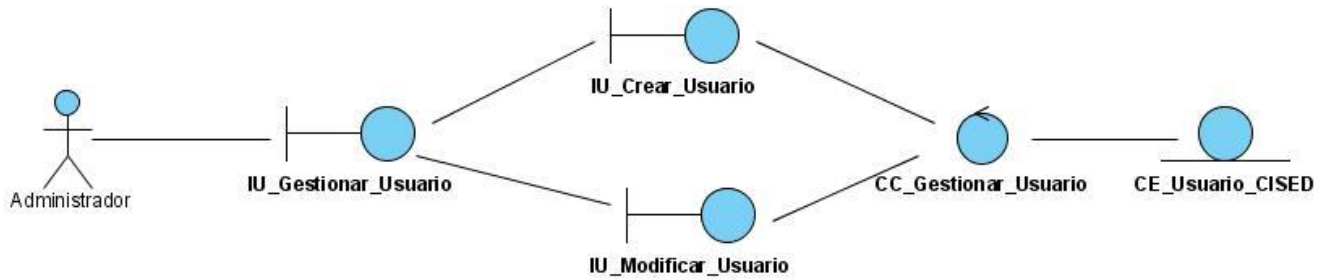


Fig. 8 DCA_Gestionar Usuario

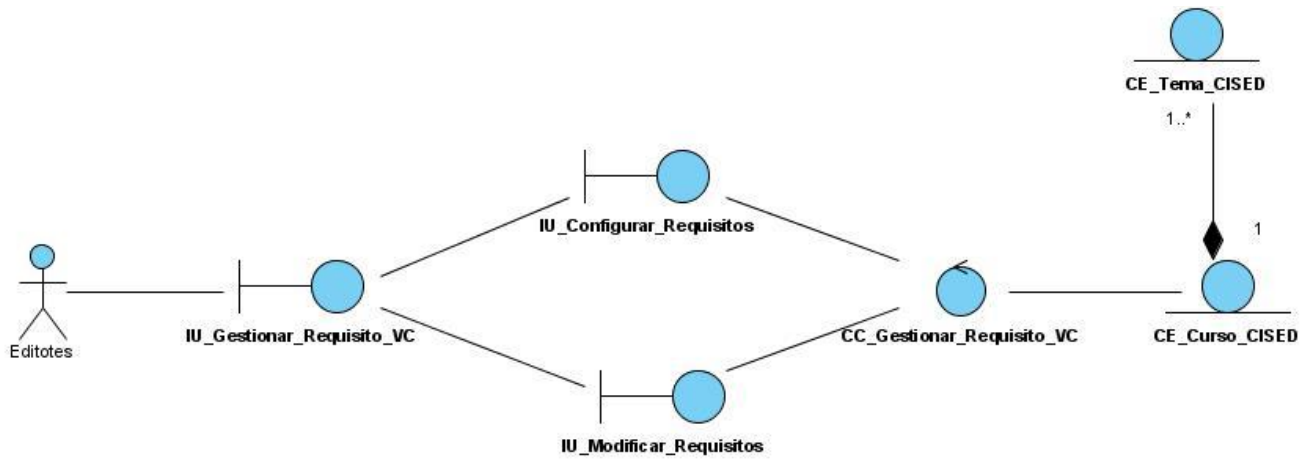


Fig. 9 DCA_Gestionar requisitos de vencimiento de cursos

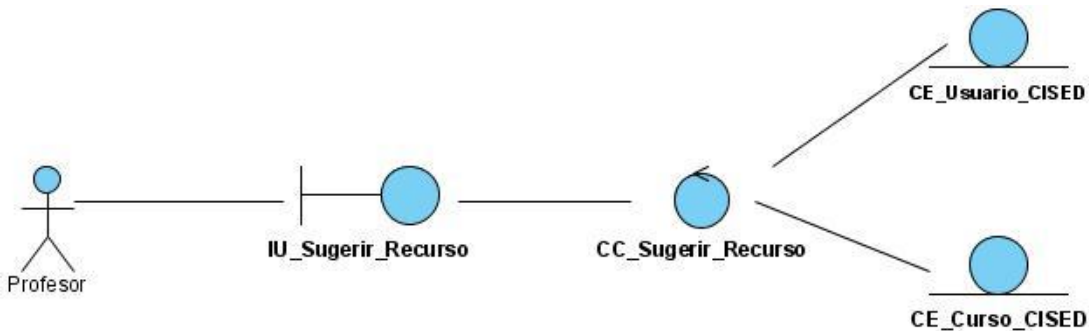


Fig. 10 DCA_Sugerir recurso

3.4.4 Diagrama de interacción de clases del análisis

Ver anexo 3

3.5 Diseño

Durante el diseño modelamos el sistema y su arquitectura, capaz de soportar los requisitos del sistema. El Modelo de Diseño es el centro de atención de la fase final de la elaboración y el comienzo

de las iteraciones de construcción. Sirve de abstracción a la implementación del sistema y se utiliza como una entrada fundamental en las actividades de implementación.

3.6 Diagramas de interacción del diseño

Los diagramas de interacción (secuencia y colaboración) son un conjunto de objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos para modelar los aspectos dinámicos del sistema.

El diagrama de secuencia es uno de los que explica detalladamente la interacción entre los objetos del sistema. Representa la secuencia de acciones de un caso de uso destacando el orden temporal de los mensajes.

3.6.1 Diagramas de interacción del Diseño.

Ver anexo 4.

3.7 Diagrama de clases del diseño

El diagrama de clases del diseño es la representación gráfica de las funcionalidades del sistema. En forma de clases estereotipadas, llámese clases: el formulario, página servidora y la página cliente.

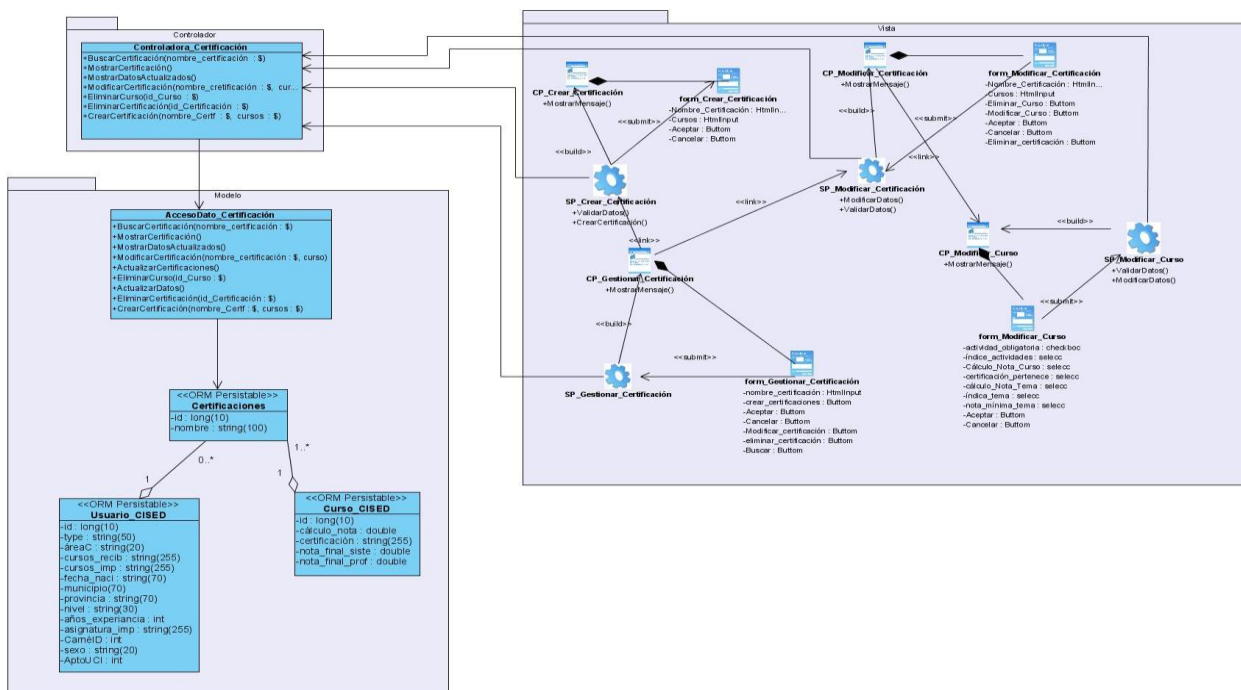


Fig. 11 CU_Gestionar certificación

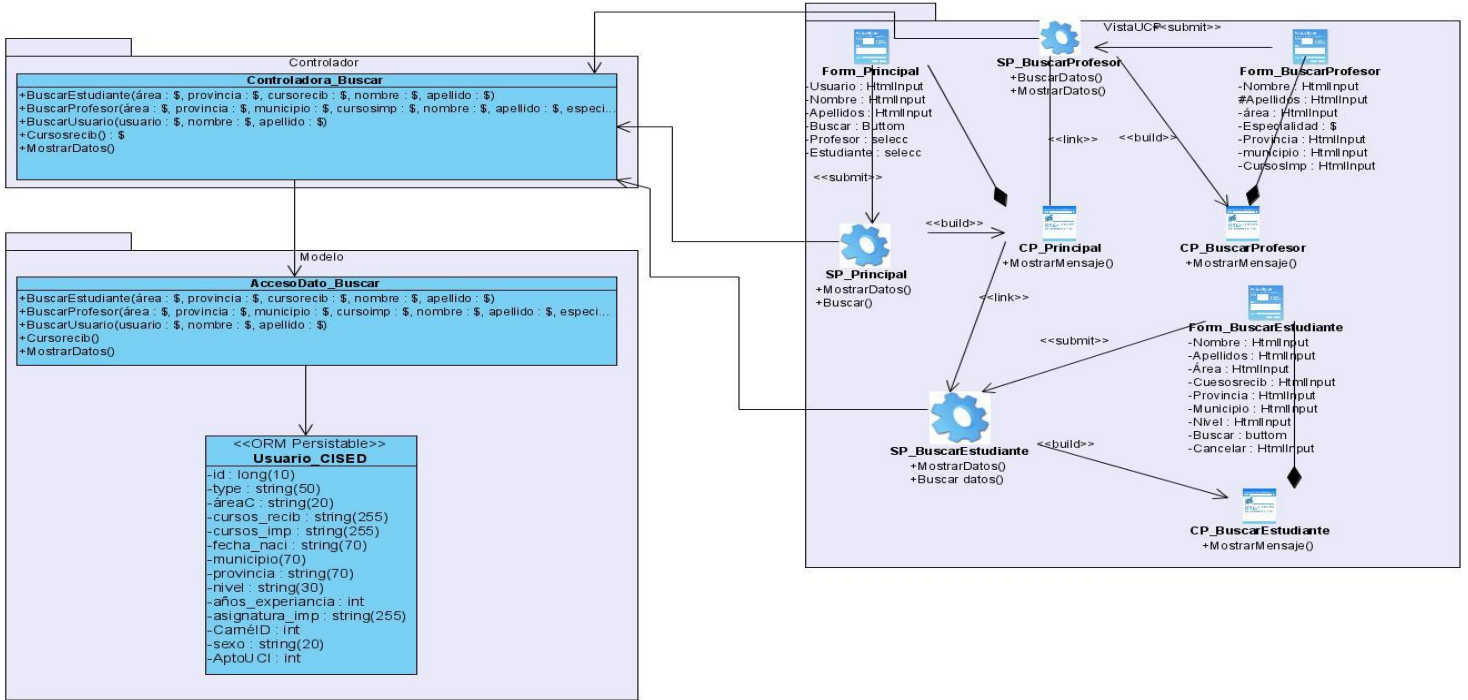


Fig. 12 CU_Buscar Usuario

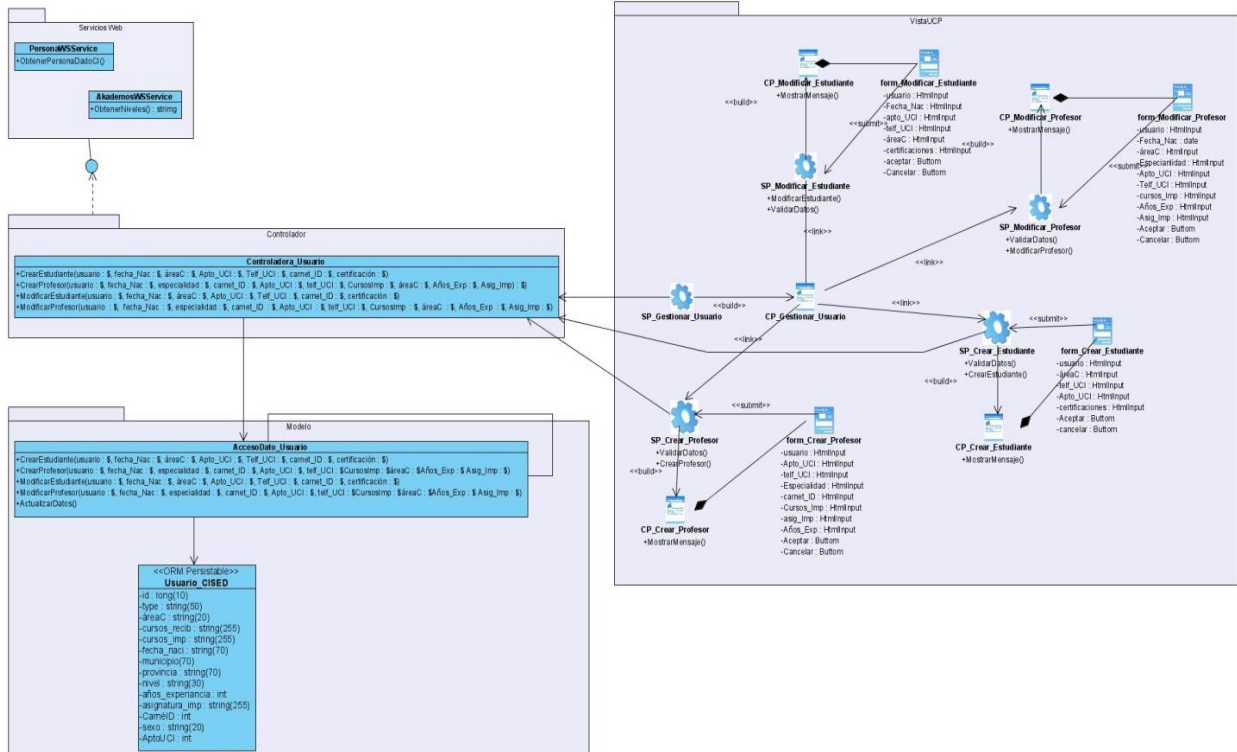


Fig. 13

CU_Gestionar Usuario

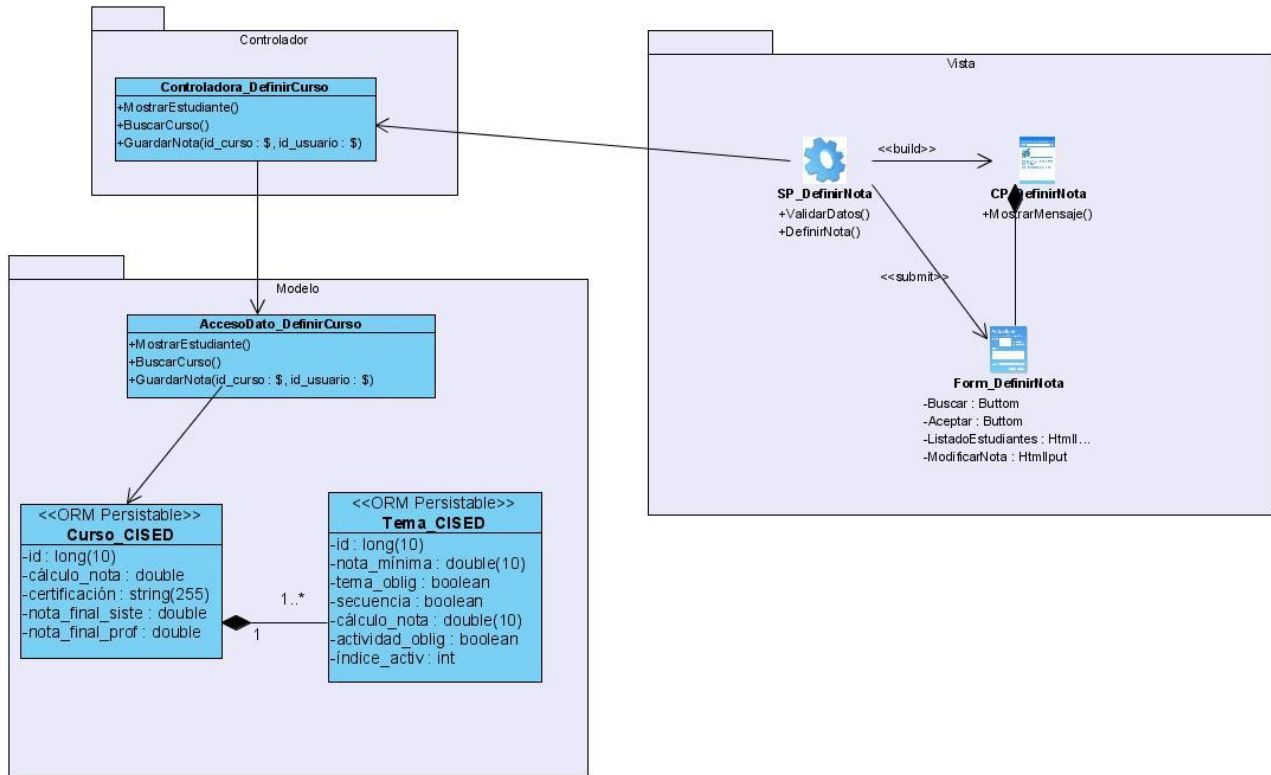


Fig. 14 Definir nota del curso

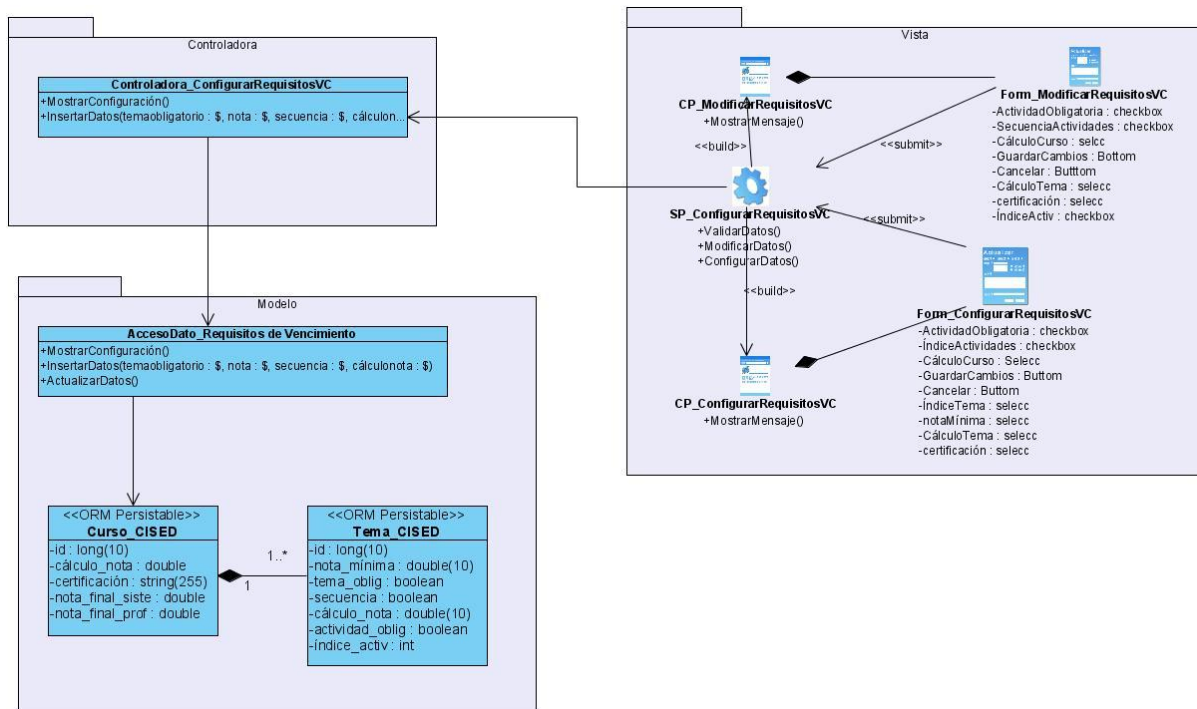


Fig. 15 Gestionar requisito de vencimiento de curso

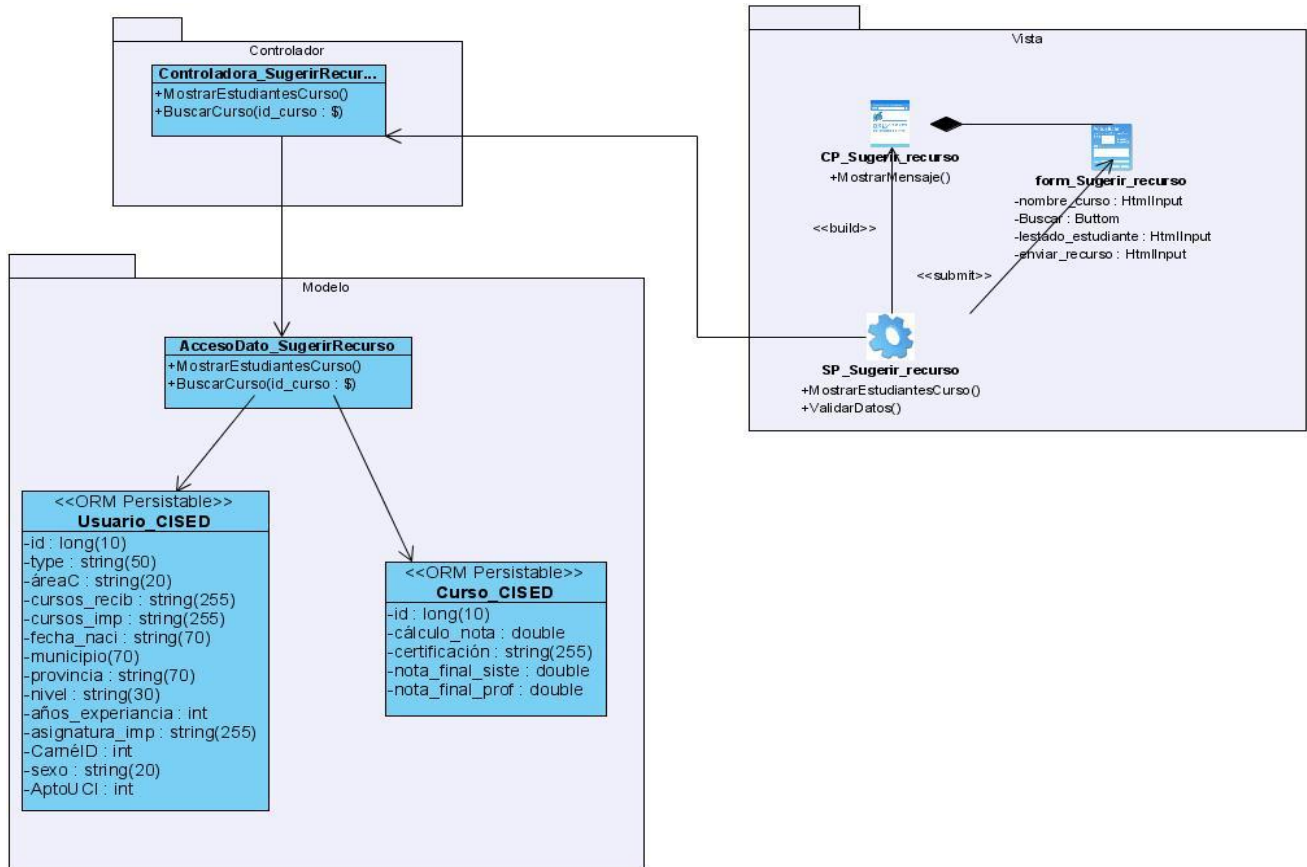


Fig. 16 CU_Sugerir recurso

3.8 Diseño de la base de datos

Al comenzar el ciclo de desarrollo de un sistema, existen aspectos importantes a tener en cuenta, uno de ellos es el diseño de la base de datos. Mediante la cual se podrán representar los datos lógicamente coherentes, con un cierto significado inherente y brindar persistencia en el modelo de diseño construido.

La plataforma Moodle posee una extensa base de datos que nos da la posibilidad de utilizar algunas de sus tablas en el sistema EVACISED y trabajar con datos que se le visualizará al usuario. En el modelo de dato del sistema se representarán todas esas tablas de Moodle y sus relaciones con la base de datos.

3.8.1 Diagrama Entidad Relación de la base de datos

En el siguiente modelo de datos se representan las tablas que almacenarán los datos persistentes en el sistema. En ellas se representan la herencia entre clases ya que se reutilizan tablas de la base de datos de Moodle, y en uno de los casos se utiliza una forma no común en los programadores al

representar la herencia, como es la representación de todos los objetos en una misma tabla. Para lograrlo, la tabla necesita tener un campo “type” que representa el tipo del objeto.

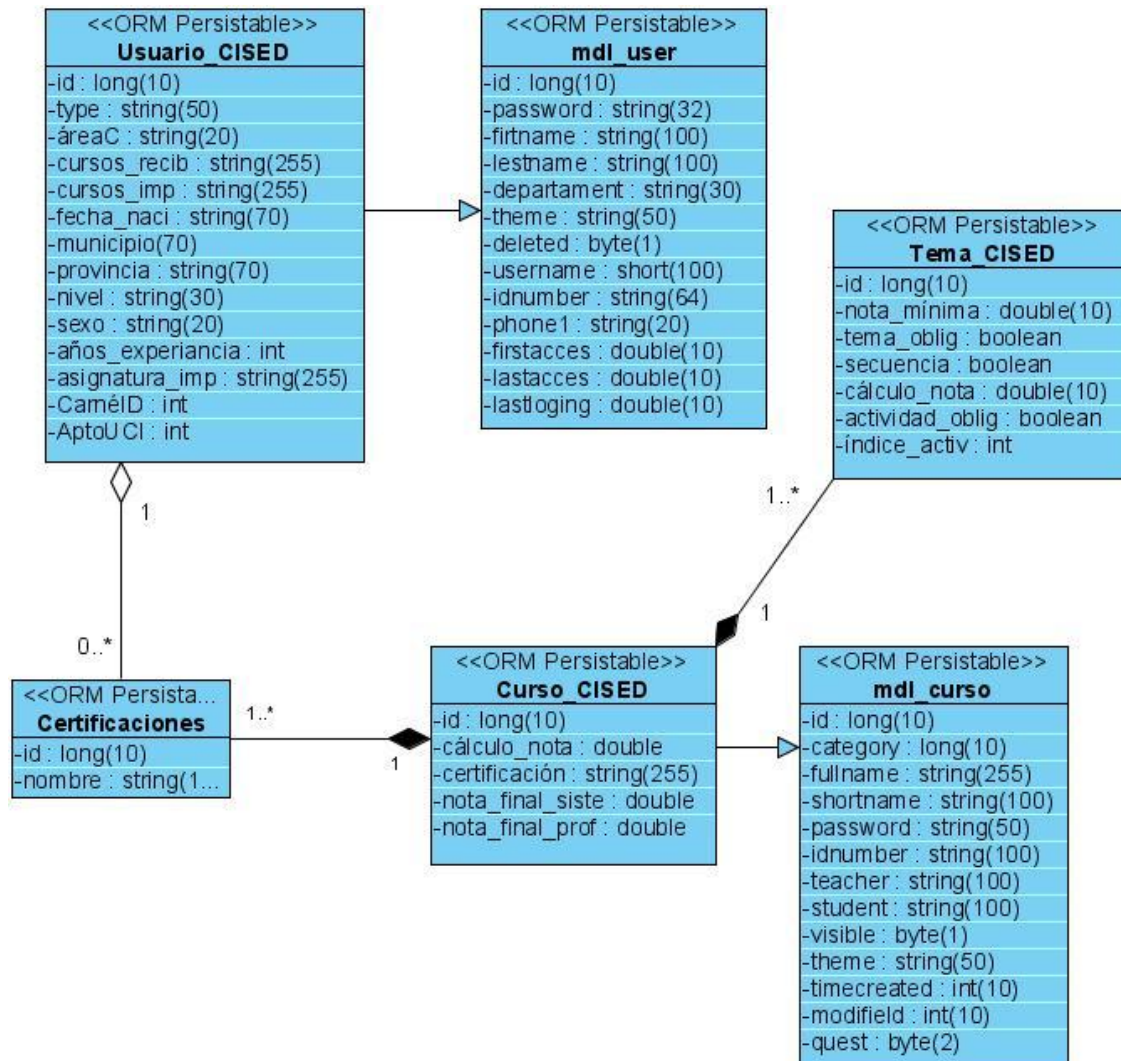


Fig. 17 Diagrama entidad relación de la base de datos

3.8.2 Diagrama de despliegue

El modelo de despliegue es la representación física de los diferentes nodos del sistema en tiempo de ejecución, y los componentes que residen en ellos. Permite la comprensión de la comunicación entre la arquitectura de software y hardware.

Un nodo constituye un procesador o un dispositivo sobre el cual se pueden desarrollar los componentes. Cada nodo debe tener un nombre específico que lo identifique de los demás.

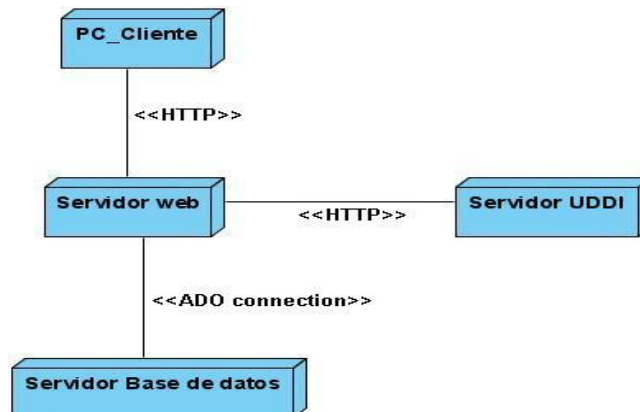


Fig. 18 Diagrama de despliegue

El diagrama establece la relación existente entre los nodos:

- **Servidor UDDI:** Servicios web que la UCI brinda para obtener información del personal que radica en la misma una vez que la persona sea autenticada en el sistema con el dominio uci.cu mediante LDAP. El protocolo que utiliza es SOAP (*Simple Object Access Protocol*) por sus siglas en inglés.
- **PC_Cliente:** Espacio de trabajo por el cual el cliente interactúa con la aplicación, posibilitando las competencias entre las personas del Centro. Esta computadora debe tener instalado el Sistema Operativo *Windows* o *GNU/ Linux*, como navegador Web *Internet Explorer* o *Mozilla firefox 2.0*.
- **Servidor Web:** Se utilizará el servidor Web apache más el lenguaje de programación PHP. Contiene la aplicación.
- **Servidor Base de dato:** Se utilizará PostgreSQL como base de dato que permitirá almacenar los datos de la aplicación.

3.9 Conclusiones

Durante el desarrollo de este capítulo se realizó la descripción y representación gráfica de los diagramas de clases y de interacción del análisis, así como los correspondientes al diseño teniendo en cuenta los patrones de diseño y el modelo arquitectónico antes propuesto. Además, se realizó el diseño del modelo de datos, el cual debe ser implementado. Una vez realizadas todas estas actividades finalmente se obtiene un diagrama de despliegue que muestra la configuración de los elementos de procesamiento y sus conexiones en el sistema, mediante nodos que representa típicamente un dispositivo o un procesador.

Conclusiones

Después de finalizar el presente trabajo de diploma se concluye que:

- Se diseñó una herramienta que se prevé sea capaz de gestionar el proceso de aprendizaje a distancia y los procesos de competencias en el CISED.
- Se realizó un profundo estudio de diferentes sistemas de gestión de contenidos educativos, seleccionándose la plataforma Moodle, la cual permite la integración con el sistema informático diseñado.
- Se elaboró el modelo de dominio del sistema el cual permitió comprender las necesidades del cliente. Así como realizar la captura de los requisitos funcionales y no funcionales del sistema.
- Se efectuó el análisis y diseño del sistema asumiendo una arquitectura y patrones de diseño que dotarán a la herramienta de robustez, seguridad y flexibilidad.
- Se logró transformar los requisitos en un diseño de software para posteriormente viabilizar la implementación del sistema.
- Se proponen prototipos no funcionales de interfaz de usuario una vez modelados los diagramas de casos de usos y sus descripciones.
- Se propone la arquitectura modelo, vista, controlador (MVC) sobre la que se diseñó el sistema de acuerdo a lo que se establece por la metodología RUP.

Recomendaciones

Para dar continuidad y seguimiento al trabajo de diploma se recomienda:

- Implementar los casos de usos del sistema propuesto.
- Incluir tablas en el modelo de datos y en el diseño en caso de ser necesario para la implementación.
- Consultar el glosario de términos como base para la comprensión de este trabajo y futuras investigaciones sobre el proceso de enseñanza a distancia.

Bibliografía

MoodleDocs. *MoodleDocs*. [Online] <http://docs.moodle.org/es/Filosof%C3%ADa>.

Agudelo, Angel Augusto. 2007. Proceso Unificado de Captura de Requisitos. [Online] 2007. [Cited: abril 5, 2007.] <http://www.scribd.com/doc/319582/Requisitos>.

Atutor. 2010. Learning Content Management System. *Learning Content Management System*. [Online] 2010. <http://www.atutor.ca/>.

Benjamín Arroyo Guerrero, Juan Fournier Torres. 2004. Creación de un espacio virtual para el intercambio de lenguaje científico. *Creación de un espacio virtual para el intercambio de lenguaje científico*. [Online] 2004. [Cited: mayo 19, 2010.] <http://www.cs.us.es/~fsancho/PFC/ficheros/CHATML.pdf>.

Consortio Claroline. 2008. Let's build knowledge together. [Online] 2008. [Cited: 22 enero, 2010.] <http://www.claroline.com>.

Download. 2007. Paradigma visual para UML. *Free Download Manager*. [Online] marzo 5, 2007. [Cited: febrero 18, 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.

Elizabeth. 2008. TECNOLOGÍA EDUCATIVA. *TECNICAS FORMATIVAS & TECNOLOGIA*. [Online] julio 3, 2008. [Cited: mayo 26, 2010.] <http://didacticaytecnologia.blogspot.com/2008/07/tecnologia-educativa-se-entiende-como-el.html>.

Gloria Marcela Haviria, Miguel Angel Rodriguez. 2009. Diagrama de Actividades. *Analisis y Desarrollo de Sistemas de Informacion*. [Online] febrero 23, 2009. [Cited: febrero 20, 2010.] http://miguelrodriguez40099.blogspot.com/2009_02_01_archive.html.

Higuita, Daimer. 2008. Gestión Por Competencias. *slideshare*. [Online] octubre 2008. [Cited: marzo 29, 2010.] <http://www.slideshare.net/kingdaimer/de-la-gestin-por-competencias-presentation>.

Hinostroza, Raul Rodas. 2007. Características de PHP. *LinuxCentro.Net*. [Online] febrero 22, 2007. [Cited: marzo 28, 2010.] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.

2010. Ingeniería de Software II. *Ingeniería de Software II*. [Online] 2010. <http://eva.uci.cu/course/view.php?id=259>.

Ivar Jacobson, Grady Booch, James Rumbaugh. 2000. *El Proceso unificado de desarrollo de software Madrid, Pearson Educación*. Madrid : s.n., 2000. ISBN.

- Kalú, Proyecto.** Metodología. *Proyecto Kalú. Portal para la Formación y Cooperación Internacional y Ayuda Humanitaria.* [Online] [Cited: febrero 10, 2010.] <http://proyectokalu.com/metodologia/>.
- Moodle Docs. 2006.** A favor de Moodle. *Moodle* . [Online] octubre 25, 2006. [Cited: mayo 19, 2010.] http://docs.moodle.org/es/A_favor_de_Moodle.
- MoodleDocs. 2007.** Moodle. *Filosofía.* [Online] agosto 4, 2007. [Cited: marzo 18, 2010.] <http://docs.moodle.org/es/Filosof%C3%ADa>.
- Peggy Karen Cruz Muñoz, Georgina M. Vega López. 2001.** *LA GESTIÓN POR COMPETENCIAS1.* Chile : s.n., 2001.
- Peñalvo, García.** Portal para la Formación en Cooperación Internacional y Ayuda Humanitaria. *Portal para la Formación en Cooperación Internacional y Ayuda Humanitaria.* [Online] [Cited: enero 20, 2010.] <http://proyectokalu.com/>.
- Pérez., Reisel González. 2008.** [Online] febrero 4, 2008. [Cited: marzo 28, 2010.] <http://www.monografias.com/trabajos-pdf2/sistema-gestion-base-datos-postgresql/sistema-gestion-base-datos-postgresql.pdf>.
- **2008.** Introducción al Sistema de Gestión de Base de Datos PostgreSQL. *Introducción al Sistema de Gestión de Base de Datos PostgreSQL.* [Online] febrero 4, 2008. [Cited: marzo 28, 2010.] <http://www.monografias.com/trabajos-pdf2/sistema-gestion-base-datos-postgresql/sistema-gestion-base-datos-postgresql.pdf>.
- pujadas, Dr.Marta.** *La formación integral de los trabajadores.Herramienta de prevención.* *Revista de Software Libre en la UCI. Proyecto Unicornios. 2010.* Habana : junio del 2007, 2010, Vol. 02.
- Santo, Jesús Baños. 2007.** Manual de consulta para el profesorado. *Manual de consulta para el profesorado.* [Online] octubre 2007. [Cited: enero 20, 2010.] <http://www.scribd.com/doc/3086874/Moodle18-Manual-Prof>.
- schermbeek, Mariangela Pocaterra Van. 2008.** [Online] febrero 2008. [Cited: marzo 29, 2010.] http://ddd.uab.cat/pub/trerecpro/2009/hdl_2072_41829/Treball+de+recerca.pdf.
- Universidad de las Ciencias Informáticas. 2010.** Introducción al manual de P1. *Introducción al manual de P1.* [Online] 2010. [Cited: Abril 25, 2010.] http://eva.uci.cu/file.php/580/Bibliografia/Manuales_de_la_asignatura/Cap_1.pdf.

Glosario de Términos

CASE (*Computer Aided Software Engineering*): en español Ingeniería de *Software* Asistida por Ordenador. Son aplicaciones informáticas para aumentar la productividad en el desarrollo de *software* reduciendo el coste de las mismas en términos de tiempo y capital.

CISED: Centro de Identificación y Seguridad Digital. Centro productivo para la informatización de sistemas de identificación del ciudadano.

CSS (*Cascading Style Sheets*): En español hojas con estilo de cascada, usado para definir la presentación de un documento estructurado escrito en HTML o XML.

E-learning: Sistemas de educación a distancia.

GRASP: Patrones Generales de Software para la Asignación de Responsabilidades

GNU/Linux: Es un sistema operativo tipo Unix, conocido como Linux que se distribuye bajo la Licencia Pública General de GNU (GNU GPL) de software libre.

GPL (*General Public License*): Licencia Pública General que permite el uso y modificación del código para desarrollar solo software libre.

HTTP (*Hyper Text Transfer Protocol*): Protocolo de transferencia de hipertexto usado en cada transacción de la Web.

HTML (*HyperText Markup Language*): Lenguaje de enmarcado.

IMAP, NNTP, CAS, FirstClass, Shibboleth: Protocolos de acceso, que permite la comunicación y navegación en la red.

IMS Content Packaging y SCORM: Paquetes de contenido usados en *e-Learning* para definir algunos contenidos de aprendizaje.

LAMP: Combinación de las tecnologías (Linux; Sistema operativo, Apache; servidor Web, MySQL como gestor de base de datos).

Logs: Registro de todos los hitos que un servidor ha recibido en un período de tiempo dado. Puede ser utilizado por auditores externos para registrar el uso de los sitios.

MySQL: Sistema de gestión de base de datos relacional (SGBD).

MVC: “Modelo, Vista, Controlador” es un patrón arquitectónico que permite organizar el código de una aplicación, facilitando modificar una de las capas sin necesidad de modificar las restantes, así como la posterior reutilización del código.

MVCC (Multiversion Concurrency Control): Multiversión de control de Concurrency.

MTF (Microsoft Solution Framework): Metodología de desarrollo software centrada en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

OMG (Object Management Group): Objeto de administración de grupos.

PHP: Lenguaje de programación para el desarrollo web. Es un lenguaje interpretado donde el navegador web es quien visualiza lo que el usuario desea.

PostgreSQL: Es un sistema de gestión de base de datos relacional orientada a objetos de software libre.

SOAP (Simple Object Access Protocol): Protocolo que define la forma en que dos objetos, en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

Solaris: Sistema operativo tipo Unix.

Scrum: Metodología ágil para el desarrollo de sistemas software.

SQL (Structured Query Language): Lenguaje estructurado de consulta.

UML: “Unified Modeling Language”. Lenguaje de representación gráfica a través del cual podemos especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

Unix: Es un sistema operativo portable, multitarea y multiusuario.

WCM: Sistema de Gestión de Contenido Web, implementado como una aplicación Web, para crear y administrar contenidos.

Web: Sistema para presentar información en Internet basado en hipertexto.

Wiki: Sitio web donde las páginas pueden ser editadas por los usuarios de forma tal que pueden crear, modificar o borrar un mismo texto.

WML (Wireless Markup Language): Es una versión reducida del lenguaje HTML que proporciona la conexión a Internet y la visualización de páginas web.

XML (Extensible Markup Language): siglas en español, Lenguaje de Marcas Extensible. Es un formato estándar diseñado para el intercambio de datos en la Web.