



Universidad de
las Ciencias Informáticas



Instituto Superior Politécnico
José Antonio Echevarría



Módulo de aprendizaje para el manejador de contenidos XOOPS



Autores: Darvin Rivera Aguilar
Israel Rodríguez Avilés

Tutores: Ing. Sergio Enrique Tamayo
Ing. Yunier Saborit Ramírez

julio 2005

DECLARACION DE AUTORIA

Nosotros: *Darvin Rivera Aguilar e Israel Rodríguez Avilés* nos declaramos como únicos autores de este trabajo y autorizamos la Universidad de las Ciencias Informáticas (UCI) y al Centro de Estudios de Ingeniería y Sistemas (CEIS) para que hagan el uso de este trabajo de la manera que estimen conveniente.

Y para que así conste firmamos la presente a los ____ días del mes de _____ del 2005.

Firma del Autor

Firma del Autor

Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado: “**Módulo de aprendizaje para el manejador de contenido Xoops**”, fue realizado en la Universidad de Ciencias Informáticas. Este centro considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta Universidad los beneficios siguientes:

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a _____

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Módulo de aprendizaje para el manejador de contenidos Xoops.

Autor: Darvin Rivera Aguilar e Israel Rodríguez Avilés.

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de ____.

Firma

Fecha

Agradecimientos.

Antes de todo agradecerle a la Revolución Cubana por darnos la oportunidad de forjarnos como profesionales.

A mis padres por depositar toda su confianza en mí.

A mi antigua decana de la Universidad de Camaguey por darme todo su apoyo y depositar toda su confianza en mí.

A mis compañeros de cinco años de estudio y esfuerzo, por brindarme su amistad desinteresada, y compartir tantas cosas buenas y malas, que durarán en nuestra memoria para siempre.

A mi oponente el Ing. Febe Angel Ciudad Ricardo por su gran ayuda y preocupación con nosotros.

A mis tutores el Ing. Sergio Enrique Tamayo y Yunier Saborit Ramírez por su apoyo durante la realización de este trabajo.

A todos los mencionados, y los que se me olvidaron, siéntanse parte de este resultado. Si se sienten satisfechos entonces se habrá cumplido el objetivo.

Israel Rodríguez Avilés

Agradecimientos.

Primeramente quiero agradecerle a Dios por ayudarme tanto en estos cinco años de carrera.

... a mi familia por apoyarme en lo posible y en lo imposible.

... a mis amigos Alexei y Yasser por apoyarme aún estando tan lejos.

... a mis compañeros de estudio por ayudarme en lo que pudieron, y a Yanier por aportar su penúltimo granito de arena con la portada.

... a mi oponente Febe por su ayuda y preocupación por nosotros desde el principio.

... a mis tutores el Ing. Sergio Enrique Tamayo y Yunier Saborit Ramírez por su apoyo durante la realización de este trabajo.

... a mi Decana Geisel Salgado por ayudarme en lo que pudo.

... a Eddy Diepa por aportar su granito de arena...

... a la China y a Mayte por obligarme a estudiar... aunque no siempre pudieron...

... a Marcos Witt, Danilo Montero, Marco Barrientos, Jesús Adrian Romero, Lali Torres, Lilly Goodman y muchos, pero muchos otros que me levantaron, fortalecieron, inspiraron, hablaron, acompañaron... con sus canciones a lo largo de la carrera y sobre todo en esta etapa final.

... y a Stitch por apoyarme en la etapa final de mi carrera, y por dejarme a Tito y su corazoncito para que me acompañaran mientras no estuviera...

Darvin Rivera Aguilar

Dedicatoria

Este trabajo esta dedicado a nuestros padres que fueron ellos quienes con más sacrificio lograron que llegáramos hasta aquí y a todas las personas en general que un día aportaron su granito de arena por brindarnos su ayuda.

También a aquellas que nos quieren tanto.

"Para cambiar es necesario saber; para saber hay que aprender; y para aprender hay que hacer grandes sacrificios."

Samuel Aun Weor

Resumen

La educación a distancia puede ser considerada como una novedosa estrategia educativa que se sustenta en el uso racional y adecuado de las nuevas tecnologías, estructuras operativas flexibles y métodos pedagógicos que resultan altamente eficientes en el proceso enseñanza-aprendizaje y que la hace, hoy en día, vital en la formación del profesional de cualquier rama. El alumno de los sistemas no presenciales goza de una serie de beneficios tales como flexibilidad de días y horarios de estudio, ejercitación y evaluación y acceso a los contenidos y a las tutorías desde el lugar en donde él decida. Por otro lado en los últimos años se ha desarrollado el concepto de sistema de gestión de contenidos (Content Management Systems o CMS), los cuales son herramientas que proporcionan un entorno donde la actualización, mantenimiento y ampliación de portales Web con la colaboración de múltiples usuarios es sencilla.

Este trabajo propone la creación de un módulo del Sistema de Gestión de Contenido XOOPS que permita el montaje de cursos a distancia dentro del portal de Software Libre de la UCI. Con el mismo se garantizará que la comunidad de usuarios del centro tenga un punto de referencia para el aprendizaje de estas nuevas tecnologías de estratégica importancia para el desarrollo del país.

En los primeros capítulos se describen las tecnologías utilizadas, dando una panorámica general del estado del arte y la tendencia actual de las mismas. Seguidamente se explica detalladamente la propuesta de solución con cada uno de sus modelos y diagramas. Por último el estudio de factibilidad del sistema da una visión general de los costos y beneficios de la implementación de este proyecto.

Índice

INTRODUCCIÓN	1
CAPÍTULO I.	5
FUNDAMENTACIÓN DEL TEMA.	5
1.1. INTRODUCCIÓN.	5
1.2. ESTADO DEL ARTE.	5
1.3. SISTEMA DE GESTIÓN DE CONTENIDO (CMS).....	7
1.3.1. <i>¿Qué es un Sistemas de Gestión de Contenidos (CMS)?</i>	8
1.3.2. <i>Creación de contenido.</i>	8
1.3.3. <i>Gestión de contenido.</i>	8
1.3.4. <i>Publicación de contenido.</i>	9
1.3.5. <i>Presentación de contenido.</i>	9
1.3.6. <i>CMS comerciales y de código abierto.</i>	10
1.3.7. <i>Los CMS en el e-Learning.</i>	11
1.4. SISTEMA DE GESTIÓN DE APRENDIZAJE (LMS).....	12
1.4.1. <i>¿Qué es un Sistema de Gestión de Aprendizaje (LMS)?</i>	13
1.4.2. <i>LMS comerciales y de código abierto.</i>	16
1.5. <i>¿QUÉ ES UN E-LEARNING?</i>	16
1.6. <i>¿QUÉ ES UN SISTEMA DE GESTIÓN DE CONTENIDOS EDUCATIVOS (LCMS)?</i>	17
1.6.1. <i>Diferencia entre un LMS y un LCMS.</i>	18
1.6.2. <i>Característica comparativa entre LMS y LCMS.</i>	20
1.7. OBJETO DE ESTUDIO.....	20
1.7.1. <i>Descripción general.</i>	20
1.7.2. <i>Ubicación.</i>	21
1.7.3. <i>Situación problémica.</i>	21
1.7.4. <i>Problema.</i>	21
1.8. OBJETIVO DEL TRABAJO.....	21
1.8.1. <i>Objetivo general.</i>	21
1.8.2. <i>Objetivos específicos.</i>	21
1.9. CONCLUSIONES.	22
CAPÍTULO II.	23
TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR.	23
2.1. INTRODUCCIÓN.	23
2.2. FUNDAMENTACIÓN DE LA METODOLOGÍA A UTILIZAR.....	23
2.2.1. <i>El proceso unificado del software (RUP).</i>	23
2.2.2. <i>El Lenguaje de Modelamiento Unificado (UML).</i>	24
2.3. FUNDAMENTACIÓN DE LA TECNOLOGÍA A UTILIZAR.	26
2.3.1. <i>PHP.</i>	26
2.3.2. <i>¿Por qué utilizar PHP?</i>	27
2.3.1. <i>MySQL</i>	29
2.3.2. <i>¿Por qué usar MySQL?</i>	29
2.3.3. <i>XOOPS.</i>	30
2.3.4. <i>Zend Studio.</i>	31

2.4. OTRAS HERRAMIENTAS A UTILIZAR.....	32
2.5. CONCLUSIONES.....	34
CAPÍTULO III.....	35
DESCRIPCIÓN DE LA PROPUESTA ACTUAL.....	35
3.1. INTRODUCCIÓN.....	35
3.2. MODELO DE DOMINIO.....	35
3.2.1. <i>Glosario de términos:</i>	35
3.2.2. <i>Modelo de objeto del dominio</i>	37
3.3 REQUISITOS FUNCIONALES.....	37
3.4 REQUISITOS NO FUNCIONALES.....	39
3.5 DESCRIPCIÓN DEL SISTEMA PROPUESTA.....	40
3.5.1 <i>Definición de los actores del sistema a automatizar.</i>	41
3.5.2 <i>Definición de los casos de uso del sistema.</i>	42
3.6. CONCLUSIONES.....	48
CAPÍTULO IV.....	50
CONSTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	50
4.1 INTRODUCCIÓN.....	50
4.2. DIAGRAMA DE CLASES.....	50
4.2.1. <i>Paquete 1: Perfil de Usuario.</i>	53
4.2.1.1 <i>Sub-Paquete 1.1: Cursar Curso.</i>	54
4.2.1.2 <i>Sub-Paquete 1.2: Solicitar Estado del Curso.</i>	54
4.2.1.3 <i>Sub-Paquete 1.3: Solicitar Matricular Curso</i>	55
4.2.2. <i>Paquete 2: Perfil Profesor Virtual.</i>	55
4.2.2.1 <i>Sub-Paquete 2.1: Modificar curso</i>	56
4.2.2.2 <i>Sub-Paquete 2.1: Crea Curso.</i>	56
4.2.3. <i>Paquete 3 Perfil Administración.</i>	57
4.2.3.1 <i>Sub-Paquete 3.1: Matricular Usuario</i>	58
4.2.3.2 <i>Sub-Paquete 3.2: Gestionar Profesores virtuales.</i>	58
4.3. MODELO DE DATOS.....	59
4.3.1. <i>Diagrama de clases persistentes.</i>	60
4.3.2. <i>Diagrama de Datos.</i>	61
4.4. PRINCIPIOS DEL DISEÑO, MANEJO DE ERRORES Y ESTÁNDARES DE CODIFICACIÓN...	62
4.4.1 <i>Principios de Diseño.</i>	62
4.4.2 <i>Manejo de Errores.</i>	63
4.4.3 <i>Estándares de Codificación.</i>	63
4.5. <i>Diagrama de despliegue.</i>	64
4.6. DIAGRAMA DE COMPONENTES.....	65
4.7. CONCLUSIONES.....	78
CAPÍTULO V.....	79
ESTUDIO DE FACTIBILIDAD.....	79
5.1. INTRODUCCIÓN.....	79
5.2. PLANIFICACIÓN.....	79
5.3. COSTOS.....	81

5.4. ANÁLISIS DE COSTOS Y BENEFICIOS.	83
5.5. CONCLUSIONES.	84
CONCLUSIONES GENERALES.....	85
RECOMENDACIONES.....	86
BIBLIOGRAFÍAS.....	87
ANEXOS.....	89
ANEXO 1	89
ANEXO 2	90
ANEXO 3	91
GLOSARIO DE TÉRMINOS.....	92

Introducción

Los avances en las tecnologías de la información y los nuevos progresos en el conocimiento de la forma que aprendemos nos permiten crear novedosos ambientes de aprendizajes para los estudiantes. Estos ambientes se caracterizan por ser interactivos, eficientes, fácilmente accesibles, flexibles, significativos y distribuidos, lo que permite que los estudiantes puedan aprender on-line o lo que lo mismo hacer de forma efectiva y real la presencia de un módulo de aprendizaje (eLearning) como una modalidad más en la actual oferta universitaria.

La Educación a distancia es un método en el que los estudiantes no necesitan asistir físicamente al aula. Normalmente, se envía al estudiante por correo el material de estudio (textos, escritos, videos, cintas de audio, *CD-Roms*) y el devuelve los ejercicios resueltos. Hoy en día, se utiliza también el correo electrónico y otras posibilidades que ofrece la introducción de las redes en los procesos de aprendizaje.

Una de las características atractivas de esta modalidad de estudios es su flexibilidad de horarios. El estudiante organiza su período de estudio por sí mismo, esto requiere cierto grado de responsabilidad por su parte.

Desde de la misma creación de la Universidad de la Ciencias Informáticas (UCI), se ha analizado la necesidad y posibilidad de poner en práctica en este centro de educación superior algunas técnicas, medios y metodologías propias de la Educación a Distancia. La masividad de estudiantes de una misma especialidad, la insuficiencia de personas altamente capacitadas y con experiencia en la educación superior, la exigencia en el proceso educativo requerida en esta Universidad y las Tecnologías Informáticas y las Comunicaciones (TIC) puesta en funcionamiento para el cumplimiento de su objeto social, creó algunas de las condiciones que posibilita que esto suceda.

Actualmente el grupo de Software Libre perteneciente a la Facultad 10 de la UCI se encuentra inmerso en varios proyectos bajo las tecnologías de código abierto (Open Source) en pos de lograr un incremento en el desarrollo de esta institución.

Con el principal objetivo de reportar a dicha facultad y a la institución acerca del estado y evolución de las tareas que le son asignadas se crea el portal de software libre, el cual incluye además diferentes módulos con diversos fines, permitiendo la actualización constante de los temas tratados.

Es responsabilidad de esta Facultad y otros grupos de investigación y desarrollo vinculados ampliar, profundizar y actualizar los conocimientos de los estudiantes inmersos en los distintos proyectos. Esta labor se quiere realizar a través de este portal, aparte de los diferentes temas de contenido que maneja se quiere realizar un pequeño módulo de e-learning e introducirlo en este sitio para montar cursos que les sirvan a los estudiantes integrados a los proyectos para su aprendizaje. Estos cursos se montaran con la tecnología que requiere un Sistema de Gestión de Aprendizaje (LMS).

En la UCI existe una cantidad creciente de estudiantes, profesores y trabajadores que se interesan por el Software Libre sin que exista la posibilidad real de poder satisfacer sus necesidades de aprendizaje con cursos presenciales debido la poca cantidad de especialistas disponibles del tema. En el caso de los proyectos que son estratégicos se realicen en software libre la situación se agudiza. Esta **situación problemática** da al traste con el desarrollo del trabajo.

El **problema** radica, que no existe actualmente en la UCI un sistema propio de gestión educativo a distancia a través de redes que partiendo de las características del proceso educativo y demás características de la institución universitaria, cumpla con los estándares internacionales en cuanto a compatibilidad y reusabilidad, que a la vez garantice la aplicación de los modernos niveles de pedagogía cubana, para lograr informatizar el proceso docente educativo en este centro de Educación Superior.

En la **actualidad** existen algunas plataformas de educación a distancia en la institución como: (SEPAD, AprendDist), que no son propia de esta organización, también existen el portal de las asignaturas de Inglés y el portal de Física pero no cumplen con todas las características y los objetivos de un modulo de enseñanza a distancia que pueda ser insertado en cualquier sitio Web. Esto crea una **necesidad** a la institución universitaria de ser propietaria de una plataforma de educación a distancia adaptable a cualquier desarrollo de los mismos.

Dentro del **objeto de estudio** del trabajo se encuentra todo lo relacionado con los Sistemas de Gestión de Contenido y el módulo de aprendizaje de los Sistemas de Gestión de Aprendizaje así como las características que debe tener dicho modulo de aprendizaje.

En este trabajo **el campo acción** se propone trabajar fundamentalmente sobre el estudio de los módulos e-learning de las plataformas virtuales educativas con las facilidades que les brinda a los usuarios. Implementada sobre el lenguaje PHP y MySQL como Gestor de Bases de Datos dada las facilidades de trabajo que brinda.

A partir de la situación problémica se desprende el **objetivo general** de este trabajo que es realizará un módulo de una plataforma de educación a distancia para insertarlo en el portal de software libre de la UCI.

Este trabajo tiene como **objetivos específico** realizar una herramienta que brinde a los estudiantes diferentes cursos, hacer una herramienta que permita la edición de cursos y una herramienta de chequeo de los cursos que se quieran impartir para mostrarlo a los clientes.

Para cumplir estos objetivos se realizaron las siguientes tareas:

- Hacer un estudio sobre el entorno de trabajo.
- Hacer un estudio de las necesidades que presenta el cliente.

-
- Hacer un estudio de los LMS y las características que presenta el módulo de aprendizaje.
 - Hacer un estudio de los CMS y sus funcionalidades.
 - Analizar los requisitos que debe cumplir el sistema.
 - Seleccionar la metodología de análisis y diseño del sistema que garantice su creación y calidad.
 - Seleccionar las herramientas para llevar a cabo el proyecto a desarrollar.

El presente documento está estructurado por cinco capítulos:

Capítulo 1: Se realiza un estudio del estado del arte y los fundamentos teóricos que brinda argumentos básicos que son necesarios para el desarrollo de los temas a tratar en el transcurso del documento.

Capítulo 2: Se describen las tecnologías actuales y diferentes herramientas a considerar para el desarrollo de la propuesta, analizándose la metodología de desarrollo del sistema.

Capítulo 3: Se realiza un análisis del modelo de dominio del sistema a desarrollar, se describe la solución propuesta utilizando los requerimientos funcionales y no funcionales, los casos de usos y el diagrama de casos de usos, los actores y trabajadores del sistema.

Capítulo 4: Se describe la solución a través de los diagramas de clases del diseño y el de clases persistentes, se define además los principios de diseño del sistema seguidos en la aplicación.

Capítulo 5: Se describe el estudio de factibilidad realizado para este proyecto y se analiza su factibilidad en cuanto al desarrollo del software. A continuación se encontrarán las páginas de conclusiones, la bibliografía y el glosario de términos.

Capítulo I.

Fundamentación del tema.

1.1. Introducción.

En este capítulo se hace una visión global hacia los fundamentos teóricos y temas relacionados con los Sistemas de Gestión de Contenido y los Sistemas de Gestión de Aprendizaje, estableciendo como métrica fundamental la necesidad de la utilización de esta estructura para la formación de estudiantes a través de las redes.

1.2. Estado del Arte.

Desde el triunfo de la Revolución en 1959, se realizaron varias transformaciones en diferentes esferas, siendo una de las priorizadas la de la educación. Para dar respuesta a esta necesidad objetiva de una parte de la población, el Ministerio de Educación Superior acometió la tarea de organizar una nueva modalidad de enseñanza en el país, que conjugara adecuadamente las disponibilidades en instalaciones y recursos humanos con las ansias de estudio de nuestro pueblo. Así surgió la Enseñanza Dirigida y Educación a Distancia en Cuba en el Curso 1979–80, coincidiendo con la década de auge de esta modalidad a nivel mundial.

Esta nueva modalidad de estudios no surgió como una institución independiente de las existentes, sino que se insertó en la estructura organizativa de la educación presencial, y se fue desarrollando a los que con el tiempo se le fue incorporado, en forma experimental, otros medios auxiliares como el video, la radio, el audio casete y, así sucesivamente, sin considerar hasta el momento el contacto directo con el profesor.

Hoy con el avance tecnológico, el surgimiento de las redes y la introducción de Internet e intranet en el país se han introducido nuevas tecnologías para la adecuación

a través de la red, que son las llamadas plataformas de cursos a distancias, con el objetivo de ampliar más el área de educación sin el estado presencial del estudiante y del profesor. Se ha pretendido desarrollar una herramienta capaz de brindar las mayores facilidades para educar sin que el estudiante sea afectado en sus tareas extra docentes donde los estudiantes podrán seguir los cursos impartidos, realizar los trabajos encomendados y auto evaluaciones para conocer el progreso en la asignatura, participar en foros electrónicos sobre su contenido, publicar mensajes en tableros de anuncios, contactar con el profesor a través de correo electrónico, todos ellos supervisado por un instructor (profesor), encargado de velar por el buen funcionamiento de las clases que está impartiendo on-line.

Los profesores podrán publicar los contenidos del curso, realizar evaluaciones a los alumnos, controlar el acceso y cada parte concreta del curso, asignar tareas, comunicarse con los alumnos de forma individual y colectiva, así como muchas otras funciones de control y evaluación.

En la actualidad podemos decir que los LMS están saliendo de su estado embrionario, evolucionando muy rápidamente, aunque muy fragmentado, es decir, muchas diferentes perspectivas de otras tantas organizaciones intentando posicionar sus soluciones. El término LMS se presenta al mercado como la evolución natural dentro de la industria de la formación y la educación y como la siguiente fase dentro de la revolución digital.

La Educación a Distancia está siendo cada vez más completa y sofisticada y son ya muchos los elementos que se necesitan por parte de los expertos y que han venido a enriquecer las posibilidades de esta modalidad de aprendizaje, como puede ser, por ejemplo, el libro electrónico que, por sus enormes posibilidades didácticas, se está abriendo camino con fuerza en este nuevo escenario de enseñar y aprender.

En el país se han implantado ya varias plataformas a distancia como el SEPAD, el AprenDIST y otras, con fines de introducir algunas técnicas, medios y metodología de la educación a distancia para influir en desarrollo de la formación de estudiantes. Este un proceso tan evolutivo que se está haciendo un análisis muy profundo para seguir

llevando este proyecto a todos los centros educativos y que el mismo cumpla con todos los requisitos requeridos.

1.3. Sistema de Gestión de Contenido (CMS).

Desde hace algún tiempo se han ido desarrollando lo que se conoce como Sistemas de Gestión de Contenidos (SGC) o *Content Management System (CMS)*, estos sistemas están destinados a los usuarios que deseen construir aplicaciones Web dinámicas con rapidez y obtener a la vez resultados de alta calidad. Su gran diferencia radica en permitir que el contenido de su sitio sea modificado de forma rápida y segura desde cualquier computadora conectada al Internet. [1]

El sistema consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo público. [2]

Son dentro de las plataformas de e-Learning las más básicas, se caracterizan por no poseer herramientas elaboradas de colaboración (foros, chat, weblog, etc.) ni soporte en tiempo real. Suelen llamarse Authorware y los centros funcionales generalmente son Cursos, Grupos de Cursos, Alumnos y Grupos de Alumnos. No es posible, gestionar correlatividades, PRE-requisitos, planes de carrera, evaluación de relaciones, etc. Se utilizan comúnmente en proyectos verticales, cuando la organización cliente no posee su administrador de Aprendizaje, donde es necesario capacitar a un grupo en contenidos específicos en un tiempo muy corto. Son fácilmente implementables y de bajo costo. [17]

1.3.1. ¿Qué es un Sistemas de Gestión de Contenidos (CMS)?

El Sistemas de Gestión de Contenidos es un software que se utiliza principalmente para facilitar la gestión de Web, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido Web (*Web Content Management* o WCM). Hay que tener en cuenta, sin embargo, que la aplicación de los CMS no se limita sólo a las Web, existen diferentes divisiones acerca de la funcionalidad de los Sistemas de Gestión de Contenidos como: creación de contenido, gestión de contenido, publicación y presentación. [1] (Ver **Anexo 1**)

1.3.2. Creación de contenido.

Un CMS aporta herramientas para que los creadores sin conocimientos técnicos en páginas Web puedan concentrarse en el contenido. Lo más habitual es proporcionar un editor de texto WYSIWYG (What You See Is What You Get, que en español sería: Lo Que Ves Es Lo Que Obtienes), en el que el usuario ve el resultado final mientras escribe, al estilo de los editores comerciales, pero con un rango de formatos de texto limitado. Esta limitación tiene sentido, ya que el objetivo es que el creador pueda poner énfasis en algunos puntos, pero sin modificar mucho el estilo general del sitio Web. Un CMS siempre tendría que proporcionar un editor WYSIWYG por su facilidad de uso y la comodidad de acceso desde cualquier ordenador con un navegador y acceso a Internet. [1]

Para la creación del sitio propiamente dicho, los CMS aportan herramientas para definir la estructura, el formato de las páginas, el aspecto visual, uso de patrones, y un sistema modular que permite incluir funciones no previstas originalmente. [1]

1.3.3. Gestión de contenido.

Los documentos creados se depositan en una base de datos central donde también se guardan el resto de datos de la Web, cómo son los datos relativos a los documentos (versiones hechas, autor, fecha de publicación y caducidad, etc.), datos y preferencias de los usuarios, la estructura de la Web, etc. [1]

La estructura de la Web se puede configurar con una herramienta que, habitualmente, presenta una visión jerárquica del sitio y permite modificaciones. Mediante esta estructura se puede asignar un grupo a cada área, con responsables, editores, autores y usuarios con diferentes permisos. Eso es imprescindible para facilitar el flujo de trabajo (*workflow*) con un circuito de edición que va desde el autor hasta el responsable final de la publicación. El CMS permite la comunicación entre los miembros del grupo y hace un seguimiento del estado de cada paso del flujo de trabajo. [1]

1.3.4. Publicación de contenido.

Una página aprobada se publica automáticamente cuando llega la fecha de publicación, y cuando caduca se archiva para futuras referencias. En su publicación se aplica el patrón definido para toda la Web o para la sección concreta donde está situada, de forma que el resultado final es un sitio Web con un aspecto consistente en todas sus páginas. Esta separación entre contenido y forma permite que se pueda modificar el aspecto visual de un sitio Web sin afectar a los documentos ya creados y libera a los autores de preocuparse por el diseño final de sus páginas. [1]

1.3.5. Presentación de contenido.

Un CMS puede gestionar automáticamente la accesibilidad del Web, con soporte de normas internacionales de accesibilidad, y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (Windows, Linux, etc.) y su capacidad de internacionalización le permite adaptarse al idioma del visitante.

El sistema se encarga de gestionar muchos otros aspectos como son los menús de navegación o la jerarquía de la página actual dentro del Web, añadiendo enlaces de forma automática. También gestiona todos los módulos, internos o externos, que incorpore al sistema. Así por ejemplo, con un módulo de noticias se presentarían las novedades aparecidas en otro Web, con un módulo de publicidad se mostraría un

anuncio o mensaje animado, y con un módulo de foro se podría mostrar, en la página principal, el título de los últimos mensajes recibidos. Todo eso con los enlaces correspondientes y, evidentemente, siguiendo el patrón que los diseñadores hayan creado. [1]

1.3.6. CMS comerciales y de código abierto.

Se puede hacer una primera división de los CMS según el tipo de licencia escogido. Por una parte están los CMS comercializados por empresa que consideran el código fuente un activo más que tienen que mantener en propiedad, y que no permiten que terceros tengan acceso. Por la otra tenemos los de código fuente abierto, desarrollados por individuos, grupos o empresas permiten el acceso libre y la modificación del código fuente.

La disponibilidad del código fuente posibilita que se hagan personalizaciones del producto, correcciones de errores y desarrollo de nuevas funciones. Este hecho es una garantía de que el producto podrá evolucionar incluso después de la desaparición del grupo o empresa creadora.

Algunas empresas también dan acceso al código, pero sólo con la adquisición de una licencia especial o después de su desaparición. Generalmente las modificaciones sólo pueden hacerlas los mismos desarrolladores, y siempre según sus prioridades.

Los CMS de código abierto son mucho más flexibles en este sentido, pero se podría considerar que la herramienta comercial será más estable y coherente al estar desarrollada por un mismo grupo. En la práctica esta ventaja no es tan grande, ya que los CMS de código abierto también están coordinados por un único grupo o por empresas, de forma similar a los comerciales.

Utilizar una herramienta de gestión de contenidos de código abierto tiene otra ventaja que hace decidirse a la mayoría de usuarios: su coste. Habitualmente todo el software de código abierto es de acceso libre, es decir, sin ningún coste en licencias.

Sólo en casos aislados se hacen distinciones entre empresas y entidades sin ánimo de lucro o particulares. En comparación, los productos comerciales pueden llegar a tener un coste que sólo una gran empresa puede asumir.

En cuanto al soporte, los CMS comerciales acostumbran a dar soporte profesional, con un coste elevado en muchos casos, mientras que los de código abierto se basan más en las comunidades de usuarios que comparten información y solución a los problemas. Las formas de soporte se pueden mezclar, y así encontramos CMS de código abierto con empresas que ofrecen servicios de valor añadido y con activas comunidades de usuarios. En el caso comercial también sucede, pero el coste de las licencias hace que el gran público se decante por otras opciones y por lo tanto las comunidades de soporte son más pequeñas. Un problema que acostumbra a tener el software de código abierto es la documentación, generalmente escasa, dirigida a usuarios técnicos. Este problema se agrava en el caso de los módulos desarrollados por terceros, que no siempre incorporan las instrucciones de su funcionamiento de forma completa y entendible.

En el mercado hay CMS de calidad tanto comerciales como de código abierto. Muchos CMS de código abierto están poco elaborados (aunque en plena evolución), pero también lo encontramos entre los comerciales. En definitiva, un buen CMS de código abierto es mucho más económico que su homólogo comercial, con la ventaja de disponer de todo el código fuente y de una extensa comunidad de usuarios. [1]

1.3.7. Los CMS en el e-Learning.

El e-learning tiene unas necesidades específicas que un CMS general no siempre cubre, o si lo hace, no da las mismas facilidades que una herramienta creada específicamente por esta función.

En general, los sistemas de gestión del aprendizaje (*Learning Management Systems* o LMS) facilitan la interacción entre los profesores y los estudiantes, aportan herramientas para la gestión de contenidos académicos y permiten el seguimiento y

la valoración de los estudiantes. Es decir, facilitan una translación del modelo real en el mundo virtual.

Un buen ejemplo de sistema de gestión de cursos es Moodle uno de los más conocidos con licencia de código abierto. Sus características pueden servir para concretar algunas de las funcionalidades que se esperan de este tipo de herramientas:

Administración de profesores y alumnos.

- Aulas virtuales que contienen toda la información de un curso y permiten la comunicación con foros o con Chat.
- Creación, mantenimiento y publicación del material de un curso, con soporte de diferentes formatos, incluidos audio y vídeo.
- Talleres virtuales.
- Exámenes y tests con valoraciones.
- Trabajos con fecha de límite de entrega y aviso al profesor en caso de incumplimiento.
- Seguimiento estadístico de las acciones del estudiante.

Estos sistemas son diferentes a los CMS, tanto por el objetivo como por las características, pero actualmente empiezan a incluir capacidades de los sistemas de gestión de contenidos. Con la integración de las dos herramientas nace un nuevo concepto, los LCMS (*Learning Content Management Systems* o sistemas de gestión de contenidos para el aprendizaje). [1]

1.4. Sistema de Gestión de Aprendizaje (LMS).

Hoy en día todavía existe una cierta confusión acerca de qué se entiende por plataforma de gestión del conocimiento o Learning Management System (LMS), ya que con relativa frecuencia se denominan así a una serie de aplicaciones mínimas

utilizadas para la enseñanza por Internet y que incluyen herramientas poco integradas entre sí.

Es el componente virtual de la educación tradicional, es un software que provee a los docentes y alumnos de funciones administrativas y académicas de la capacitación. Ellos pueden comunicarse, transferir información, evaluar y ser evaluados. No son estrictamente estándares, son más bien modelos de sistemas e interfaces integrados, no aislados, y están basados en estándares abiertos y no propietarios. [17]

Un LMS tiene como funcionalidad principal la administración del conocimiento de los cursos que una organización en particular colocada en línea para que su personal se capacite. Un LMS es capaz de asignar cursos a determinadas áreas, secciones, puestos, empleados, grupos de empleados, etc. de cualquier organización. [4]

La mayoría de los LMS están hoy en día basados en la plataforma Web, lo que implica una arquitectura cliente-servidor, donde el cliente es el navegador Web del usuario y la aplicación servidor puede ser un conjunto de páginas dinámicas que se encuentran en un servidor Web como Apache o Microsoft IIS.

1.4.1. ¿Qué es un Sistema de Gestión de Aprendizaje (LMS)?

Es una plataforma de formación con herramientas integradas que se utilizan para la creación, gestión y distribución de formación a través de la Web es decir, son aplicaciones para la creación de entornos de enseñanza-aprendizaje, integrando materiales didácticos y herramientas de comunicación, colaboración y gestión educativas. [6]

El LMS es el sistema nervioso de un ambiente de aprendizaje virtual; está diseñado para proveer la infraestructura que permita cumplir los requerimientos de los segmentos del aprendizaje virtual, y para administrar las interfaces hacia un sistema de administración de contenido de aprendizaje (LCMS, por sus siglas en inglés), redes de entrega de contenido. [6] (Ver **Anexo 2**)

El LMS es un nuevo concepto educativo que funciona a través de una computadora, conectada a Internet o sobre una plataforma de Intranet. Es una herramienta de comunicación que integra los conceptos de la educación tradicional en un sistema más rápido, efectivo y de menor costo operativo.

El LMS se basa en la interacción del usuario con el material, ya que participa activamente durante todo el desarrollo. El resultado es un mayor manejo de multimedia que incluye sonido, video, gráficos animados y modernos métodos pedagógicos que actúan todos juntos en un solo campo.

Es la estrategia más poderosa para ampliar, mejorar y actualizar las posibilidades del capital intelectual de una organización. Pero lo cierto es que el aprendizaje electrónico es tan importante como exigente. Necesita infraestructura tecnológica, una buena gestión del conocimiento interno, y también debe ofrecerse en el momento justo y con el contenido adecuado para cada uno de los destinatarios.

Excelente herramienta para distribuir notas a grupos estudiantiles, e intercambio de conocimientos y opiniones en grupos formados por estudiantes y profesores.

Es el núcleo alrededor del que giran los demás elementos. Básicamente se trata de un software para servidores de Internet/Intranet que se ocupa de:

- Gestionar los usuarios, altas, bajas, control de sus aprendizaje e historial, generación de informes.
- Gestionar y lanzar los cursos, realizando un registro de la actividad del usuario: tanto los resultados de los test y evaluaciones que realice como de los tiempos y accesos al material formativo.
- Gestionar los servicios de comunicación que son el apoyo al material online, foros de discusión, charlas, videoconferencia... programarlos y ofrecerlos conforme sean necesarios.

Es la herramienta que nos permite realizar el proceso de formación continua. La sociedad requiere una formación permanente ya que los conocimientos tienen un periodo corto de validez. Permite la variación de diferentes fuentes de información. El proceso de modernización permite que el elemento encuentre el ámbito propio entre la enseñanza moderna y la tradicional. No obstante para crear un ambiente de LMS hay que tener en cuenta la tecnología. Ello implica unos procesos de cambio de la percepción de los métodos de aprendizaje.

Principales Ventajas de los LMS:

- Ahorro en desplazamiento, tiempo y recuso ello, fomentando ahorro de costes, mejo acceso de personas desfavorecidas y accesibilidad a diversos profesores.
- Decisión trabajo sincrónico/asincrónico, el alumno puede decidir cuando y como estudia, se fomenta la flexibilidad horaria y motivación.
- Amplio uso de herramientas tecnológicas, ventajas que la tecnología intrínsecamente ofrece.
- Carácter distribuido de los recursos, fomenta el aprendizaje más amplio.

Los sistemas LMS o plataformas de e-Learning generalmente incorporan las siguientes funciones:

- Gestión y registro de cursos y alumnos.
- Control de acceso y seguimiento del progreso de los alumnos.
- Administración y programación de cursos.
- Gestión de informes.

No todas las ventajas de los LMS se observan a primera vista. Es importante estudiar cada caso, hay que enfocarse en las características propias de la institución para diseñar un plan de aplicación que explote las ventajas esenciales de la formación on-line para que facilite las ventajas no vista y los ahorros no visto a simple vista.

1.4.2. LMS comerciales y de código abierto.

El hecho de que un software se denomine "Open Source" o código abierto no implica necesariamente que sea una aplicación gratis sino que se puede ver y cambiar el código fuente (la programación). Pero sí hay muchas aplicaciones totalmente libres (gratis) y algunos están disponibles en castellano como (Claroline, Moodle, Ilias o Atutor). [5]

Como en otros ámbitos, el software libre puede constituir una solución eficaz para el desarrollo de entornos de enseñanza virtual. Su principal ventaja, más allá de su bajo coste, es la posibilidad de construir entornos abiertos, modulares, escalables, de rápida puesta en servicio, versátiles y de gran capacidad de adaptación, especialmente adecuados para comunidades amplias y dinámicas en las que se requiera un elevado grado de interacción y una respuesta ágil a las necesidades de los usuarios. [5]

Hay algunos sistemas que requieren el uso de BD Oracle u otras, aunque lo más común es la utilización de alternativas Open Source como PostgreSQL y MySQL. Más de la mitad de los sistemas emplean alguna de éstas últimas BD mencionadas. [5]

1.5. ¿Qué es un e-learning?

Es el nuevo concepto educativo que integra el uso de la tecnología y elementos didácticos, para lograr el diseño y evolución de cursos de capacitación y educación a distancia. [4]

Engloba es el desarrollo de cursos de educación y capacitación vía Internet, Intranet y CD-ROM con contenidos diversos, de acuerdo a los requerimientos específicos de cada individuo y cada organización. Así como, el análisis estratégico sobre el uso de la información que permite fusionar a los usuarios al mundo de la tecnología. [4]

El e-learning expande el conocimiento de su institución a través de una cadena de valor que incluye tanto a sus proveedores como a sus clientes. Más aún, el objetivo

del e-learning es incrementar el valor de su organización. Esto incluye la reducción de los gastos de capacitación, minimizar el tiempo de entrega de los nuevos productos dentro del mercado, incrementar la formalidad de los proveedores y aumentar el conocimiento de sus productos y su valor por parte de sus clientes. [4]

El e-learning rompe con las barreras de tiempo y espacio de la capacitación tradicional con el fin de acelerar enormemente el aprendizaje organizacional. Este aprendizaje acelerado se traduce en ahorro directo para su empresa, reduciendo el tiempo para comercializar los nuevos productos (tanto en ventas como en soporte), disminuyendo el tiempo requerido para convertir a una persona recién contratada en un empleado productivo y, finalmente, propiciando un cambio organizacional en respuesta a las fuerzas del mercado o a las depresiones económicas. [4]

E-learning ofrece:

- 100% inversión
- Ahorro ente 50% y 70% vs. la capacitación presencial.
- Acceso 7x24 y desde cualquier lugar.
- Educación dinámica y personalizada.
- Incremento en el nivel de educación del capital humano en las organizaciones.
- Creación de cursos específicos que cubren las necesidades específicas
- Nueva alternativa para la capacitación de personal. [4]

1.6. ¿Qué es un Sistema de Gestión de Contenidos Educativos (LCMS)?

Software para la gestión automatizada de cursos en línea, que incluye gestión de usuarios, de resultados y de recursos. Es un sistema de gestión de cursos con las capacidades de un CMS y por lo tanto de gestionar también los contenidos de los recursos. Aplicación de software que combina las capacidades de gestión de cursos de un Learning Management System (LMS) con las capacidades de almacenamiento y creación de contenidos de un Content Management System (CMS). [5]

Es una Mega-plataforma que incorpora la gestión de contenidos para personalizar los recursos a cada alumno. Añaden técnicas de gestión de conocimiento al modelo LMS. Son ambientes estructurados diseñados para que las organizaciones puedan implementar mejor sus procesos y prácticas con el apoyo cursos, materiales y contenidos en línea. Permiten una creación mucho más eficiente, evita redundancia y permiten administrar también la participación de diversos desarrolladores, expertos colaboradores o instructores que participan en la creación de contenidos. [17] (Ver **Anexo 3**)

Características típicas de los LCMS:

- Se basan en un modelo de "objetos de contenido".
- El contenido es reutilizable a lo largo de cursos, currículo y a lo largo toda la organizaciones.
- El contenido no está ligado necesariamente a una plantilla (template) única y se puede publicar desde diversos formatos.
- Los contenidos no están limitados a una serie de controles de navegación.
- El contenido se almacena en una base de datos centralizada.
- Los contenidos pueden localizarse por diversos criterios incluyendo diversos formatos.
- Normalmente incluyen un motor que permite adaptar el contenido a diferentes grupos de usuarios con perfiles diferentes proporcionando en algunos casos diferente ambiente o manera de visualización. [17]

1.6.1. Diferencia entre un LMS y un LCMS.

Con el desarrollo de nuevas herramientas para la administración del aprendizaje y administración de contenidos, y en virtud de que sus respectivos acrónimos son similares (LMS refiriéndose a los Learning Management Systems y LCMS (Learning Content Management System) es común confundir la funcionalidad y operatividad de estas herramientas. [17]

La principal funcionalidad de un LMS es administrar estudiantes y dar seguimiento a su aprendizaje, participación y desempeño asociados con todo tipo de actividades de capacitación. Por otro lado, un LCMS administra contenidos u objetos de conocimiento el cual busca ofrecerse a la persona indicada en el momento indicado.

[17]

Se diferencia los LMS de los LCMS en que no hay gestión de los contenidos, sino simplemente administración del curso, pero acostumbra a utilizarse como sinónimo. También conocido como Content Management System (CMS) o Virtual Learning Environment (VLE). [5]. Es importante mencionar que distinguir la diferencia no es siempre muy evidente en virtud de que algunos LCMS incluyen funcionalidades de LMS.

1.6.2. Característica comparativa entre LMS y LCMS.

	LMS	LCMS
Usuario a los que están enfocado	Gerentes y administradores de la capacitación, Instructores	Diseñadores de contenidos, diseñadores instruccionales, administradores de proyectos.
Administración	Cursos, eventos de capacitación y estudiantes.	Contenidos para el aprendizaje y de soporte al desempeño.
Administra aulas virtuales.	Si (pero no siempre)	No
Reporte del desempeño de los participantes en la lectura de estos materiales	Enfoque principal	Enfoque secundario
Colaboración entre usuarios	Si	Si
Mantiene una base de datos de usuarios	No	No
Agenda de eventos	Si	No
Herramientas para la creación de contenido.	No	Si
Organización contenido reutilizable.	No	Si
Herramientas de evaluación integrada para hacer exámenes.	Si (73% de los LMS tienen esta capacidad)	Si (92% cuentan con esta capacidad)
Herramienta flujo de trabajo work-flow	No	Si (en algunas ocasiones)

1.7. Objeto de estudio.

1.7.1. Descripción general.

La elaboración y el desarrollo de un modulo e-learning de un LMS por parte de los estudiantes o profesores lleva un profundo estudio y un análisis de el tema que se va a tratar

1.7.2. Ubicación.

El sistema se desarrolla en el departamento de software libre de la UCI que pertenece a la facultad diez. El mismo está inmerso en la elaboración de varios software bajo la tecnología código abierto.

1.7.3. Situación problemática.

Dado el poco tiempo disponible que presentan los profesores y los estudiantes envueltos en la producción de software y otras actividades le es imposible preparar a estos estudiantes presencialmente en los contenido para enfrentar los diversos problemas que deben afrontar en sus tareas asignadas por la institución, lo que se le hace necesario de una pequeña herramienta educativa fácil de acceder que le permita montar cursos en la red donde los estudiantes puedan estudiar y ser evaluado.

1.7.4. Problema.

No presenta la UCI de una herramienta propia, fácil de acceder y que cumpla con los requisitos educacionales de esta institución donde pueda montar cursos para la evaluación de los estudiantes, que por situaciones tantos los profesores como los estudiantes no pueden estar presencialmente en estos cursos.

1.8. Objetivo del trabajo.

1.8.1. Objetivo general.

Realizar un modulo e-learning de una plataforma de educación a distancia donde permita montara cursos para los estudiantes y evaluarlos a través de esto cursos.

1.8.2. Objetivos específicos.

- Permitir editar y crear los cursos en el portal de Software Libre.
- Permitir mostrar los cursos a los usuarios
- Permitir que los usuarios accedan a estos cursos.
- Permitir actualizar los cursos en el portal.
- Permitir eliminar los cursos.

-
- Permitir matricular a los usuarios en los cursos.
 - Chequear las estadísticas de los usuarios.
 - Permitir que los usuarios escojan los cursos de su interés.

1.9. Conclusiones.

En este capítulo se cateterizan las tendencias actuales y las características y soluciones que se ajustan al objeto de estudio del sistema a desarrollar. Se realiza un estudio entorno a la problemática donde se analizan las condiciones para dar solución a esta situación haciendo uso de las tecnologías y herramienta del aprendizaje a distancias.

Capítulo II.

Tendencias y tecnologías actuales a considerar.

2.1. Introducción.

En el presente capítulo se realiza un estudio de las tecnologías actuales para el desarrollo de sistemas, con el fin de utilizar la más conveniente para la construcción de la herramienta que se va a desarrollar y un análisis de la metodología a seguir.

2.2. Fundamentación de la metodología a utilizar.

2.2.1. El proceso unificado del software (RUP).

El Proceso Unificado de Desarrollo, fue creado por el mismo grupo de expertos que crearon UML, Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998. El objetivo que se perseguía con esta metodología era producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos. Como se expresaba anteriormente, esta metodología concibió desde sus inicios el uso de UML como lenguaje de modelado.

Es un proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, que parten de los casos de uso; está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

A continuación se muestran estas prácticas. [8]

- Desarrollo de software en forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente

- Verifica la calidad del software.
- Controla los cambios.

El Proceso Unificado es un proceso de desarrollo de Software. O sea es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto. El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. [9]

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos esquemas de un sistema software. De hecho, UML, es una parte esencial del Proceso Unificado – sus desarrollos fueron paralelos. [9]

No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en tres fases claves – dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. Esto es lo que hace único al Proceso Unificado.

2.2.2. El Lenguaje de Modelamiento Unificado (UML).

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Este lenguaje fue creado por un grupo de estudiosos de la Ingeniería de Software formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995. Desde entonces, se ha convertido en el estándar internacional para definir organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. Con este lenguaje, se pretende unificar las experiencias acumuladas sobre técnicas de

modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. [10]

UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes. [10]

El UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables. [9]

El UML (Lenguaje Unificado para la Construcción de Modelos) se define como un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software..." [BJR971]. "Es un sistema notacional (que, entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. [12]

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación. [13]

El UML es un lenguaje para construir modelos; no guía al desarrollador en la forma de realizar el análisis y diseño orientados a objetos ni le indica cual proceso de desarrollo adoptar. [12]

Entre sus objetivos fundamentales se encuentran: [10]

- Ser tan simple como sea posible, pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir.
- Necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son el encapsulamiento y los componentes.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

Se emplea UML porque es el lenguaje de modelado que se ha transformado en estándar los últimos años para generar los modelos de datos de las aplicaciones. UML esta basado en el paradigma orientado a objeto y posee una serie de modelos que permiten plasmar con facilidad los requerimientos de una aplicación, generar su diseño e implementarlo.

En resumen a partir del surgimiento de UML, muchas de las metodologías existentes en ese entonces, fueron adaptadas para utilizar este lenguaje, como es el caso de la Metodologías de Análisis y Diseño Orientado a Objetos (POO) tienen su historia, y han sido aplicadas en una gran variedad de industrias y problemas, por lo que pueden ser clasificadas como muy maduras.

2.3. Fundamentación de la tecnología a utilizar.

2.3.1. PHP.

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos Web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil.

El código escrito en PHP tiene la peculiaridad de ser independiente de la plataforma, esto quiere decir que una aplicación escrita en PHP puede ser ejecutada en cualquier sistema operativo, gracias a que existen interpretes de PHP para casi o todos los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. Además PHP soporta la mayoría de servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape y Planet, O'Reilly Website Pro Server, Caudium, Xitami, OmniHTTPd y muchos otros. [14]

El lenguaje PHP resulta muy atractivo y está respaldado por una gran comunidad de desarrolladores, debido a su simpleza para los programadores inexpertos, y a la vez su poder para los programadores profesionales.

2.3.2. ¿Por qué utilizar PHP?

Open Source: Significa que tiene cero costos de licenciamiento, y que existe un gran número de desarrolladores alrededor del mundo que soportan este desarrollo y que los códigos fuentes pueden ser manipulados en caso de necesidad, respetando los créditos de los desarrolladores originales. Esto garantiza la transparencia del lenguaje y permite encontrar errores de programación, seguridad, puertas traseras, código malicioso, etc., más fácilmente. Esto quiere decir que en su desarrollo hay un proceso de colaboración que hace que lo tengas inmediatamente disponibles de forma gratuita, una enorme cantidad de recursos como: el lenguaje en sí, el servidor para ejecutarlo, manuales y tutoriales sin fin, y sobre todo Scripts, Scripts y más Scripts que puedes descargar y usar en cuestión de minutos. Esta abundancia de código libremente disponible ayuda aun más en el proceso de aprendizaje.

Las aplicaciones de PHP son muy estables y no dependen de las tecnologías de los navegadores (browsers) como algunas aplicaciones en Java.

Lenguaje diseñado para desarrollar aplicaciones Web. Otras aplicaciones como ASP evolucionó de Visual Basic (VBS), y JSP y la tecnología Servlets de Java SUN provienen de este lenguaje de programación.

Curva de aprendizaje: La curva de aprendizaje de PHP resulta mucho más corta para un desarrollador que tecnologías basadas en Java o en ASP. Para el aprendizaje de PHP existen disponibles vía Web, un sinnúmero de tutoriales, manuales, códigos de ejemplo, etc. gratuitos a diferencia de otros lenguajes de programación.

Lenguaje Multiplataforma: Un Script PHP *normal* puede ejecutarse sin cambiar ni una sola línea de código en cualquier servidor que interprete PHP, es decir, en servidores Windows, Linux etc. APS, por contra, es un lenguaje propietario de Microsoft y solo puede ejecutarse en servidores Microsoft. Perl si puede ejecutarse bajo Windows y Unix, pero antes hay que realizar labores de adaptación de cada Script al concreto sistema operativo.

Integración con otros softwares. PHP tiene nativamente estructuras para soportar la integración con otros softwares sin recurrir a desarrollos de terceros ni extensiones comerciales como ASP. Específicamente, conectividad a todos los motores de bases de datos. PHP combina excelentemente con otras inmejorables herramientas, como son el servidor apache y la base de datos MySQL (MySQL o PostGreSQL), todas ellas gratuitas. ASP necesita, para algunas funcionalidades (algunas realmente básicas) acudir a programas y extensiones comerciales, de terceras empresas. Pagando, por supuesto.

Rendimiento. PHP es más rápido que ASP y que JSP. Este punto es muy controversial y al respecto se pueden encontrar pruebas de benchmarking de conclusiones opuestas. Es necesario hacer un afinamiento muy exhaustivo del código y del servicio para conseguir tiempos comparables entre las tecnologías estudiadas.

Sus limitaciones: antes de lanzar la versión 5 de PHP se hablaba como limitación que no era orientada a objetos pero ahora con esta nueva versión que es totalmente

orientada a objetos ya no se puede decir esto. Se podría decir que las limitaciones se acabaron.

Por todo lo anterior se plantea que la mejor opción PHP, se sustenta en que en este momento es sin duda alguna el lenguaje de Script más poderoso del mercado. Algo sorprendente es que el número de páginas que lo utilizan aumenta en gran medida con el tiempo, y la brecha con sus competidores se abre cada día más.

2.3.1. MySQL

Es sistema de base de datos relacional multiplataforma de código abierto, muy popular en aplicaciones Web.

MySQL es un servidor de bases de datos relacionales muy rápido y robusto. Es software libre, publicado bajo la licencia GPL (GNU Public License) y mantenido por la compañía sueca MySQL AB. Es el más indicado para aplicaciones que requieren muchas lecturas y pocas escrituras y no necesiten de características muy avanzadas, como es el caso de las aplicaciones Web. [15]

MySQL es un sistema de administración de bases de datos. Desarrollado originalmente para manejar bases de datos de gran tamaño a gran velocidad, se ha utilizado con éxito en ambientes exigentes durante años. [16]

La conectividad, la velocidad, y la seguridad hacen de MySQL una buena opción para el trabajo con bases de datos en Internet. Además se trata de un software de código abierto y de licencia libre. [16]

2.3.2. ¿Por qué usar MySQL?

Por que es MySQL el servidor de bases de datos más popular en el mundo. Es un programa libre y gratuito, muy robusto, veloz y capaz de manejar sin problemas tablas con millones de registros en sitios Web de muchas visitas. Compatibilidad con más de 20 sistemas operativos incluyendo GNU/Linux, Unix, Windows y Mac OSX. Licencia de uso sin costo y acceso al código fuente. Probada fiabilidad y rapidez. [16]

Recuerde que en el espacio incluido para MySQL puede crear todas las bases de datos que quiera y repartir el espacio entre ellas como desee. Las bases de datos pueden crearse, cambiar la contraseña de acceso y darlas de baja utilizando el panel de control. [16]

Las bases de datos de MySQL no son archivos que usted pueda subir a su sitio Web como los de Access, sino que residen en un servidor de datos separado. Por ello debe utilizar algún programa cliente que le permita conectarse al servidor de datos con el fin de crear las tablas, subir datos, editar registros, etc. Aunque MySQL está en un servidor Linux, usted puede gestionar sus bases de datos desde ordenadores con cualquier sistema operativo utilizando las aplicaciones adecuadas. [16]

2.3.3. XOOPS.

XOOPS es un sistema de administración de contenido (CMS) poderoso, flexible y fácil de usar, que está basado en el lenguaje de scripting PHP. Cuando se instala en un servidor Web provisto de MySQL, permite a los administradores manejar sitios Web dinámicos, construir comunidades en línea, gestionar usuarios, modificar a su arbitrio la maqueta del sitio y alimentar contenido a través de una interfaz sencilla. XOOPS maneja todo 'lo difícil', dejándote en libertad de concentrarte en el contenido de tu sitio. [21]

XOOPS significa eXtensible Object Oriented Portal System, es decir, Sistema extensible de portales orientado a objetos. Para ti, significa que es una herramienta construida con herramientas de largo alcance, que puede escalarse para adaptarse a tus necesidades. XOOPS se ofrece bajo los términos de la Licencia Pública General (GPL) GNU, lo cual significa que puedes usarlo y modificarlo gratuitamente. Puedes modificar y alterar el programa todo lo que necesites o quieras, en tanto te atengas a los términos de distribución de la GPL. [21]

Instalación sencilla: La instalación de XOOPS es directa y fácil de seguir. El wizard de instalación te guía por todos los pasos del proceso. Contando con los permisos y configuraciones apropiados en el servidor, sólo necesitas un nombre de usuario y una

contraseña para empezar a usar XOOPS. [21]

Impulsado por una base de datos: XOOPS utiliza una base de datos relacional para almacenar contenido, administrar datos y gestionar un sitio Web dinámico. XOOPS funciona mejor con MySQL, una base de datos relacional de código abierto disponible en muchas empresas de hospedaje Web. [21]

Diseño modular orientado a objetos: El poder de XOOPS reside en la implementación modular de sus características: está escrito siguiendo métodos de orientación a objetos. Esto permite una mejor escalabilidad, extensibilidad y personalización. [21]

Es posible agregar nuevas funciones a través de un sencillo proceso de instalación de módulos. Esta aproximación modular hace que XOOPS sea altamente personalizable. Sólo tienes que instalar los módulos que consideres útiles, y puedes retirarlos en cualquier momento --no hay necesidad de cargar tu sitio con código innecesario, y puedes configurarlo para que tenga sólo aquello que en verdad necesitas. [21]

2.3.4. Zend Studio.

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web, como no, en lenguaje PHP. [22]

El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. [22]

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un

poco más. [22]

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. [22]

2.4. Otras herramientas a utilizar.

Dreamweaver MX es un software fácil de usar que permite crear páginas Web profesionales. Las funciones de edición visual de Dreamweaver MX permiten agregar rápidamente diseño y funcionalidad a las páginas, sin la necesidad de programar manualmente el código HTML. Se puede crear tablas, editar marcos, trabajar con capas, insertar comportamientos JavaScript, etc., de una forma muy sencilla y visual. Además incluye un software de cliente FTP completo, permitiendo entre otras cosas trabajar con mapas visuales de los sitios Web. [19]

Para estas funciones la elección no ha sido muy difícil, ya que la herramienta de creación de sitios Web más utilizada en la actualidad es el Macromedia Dreamweaver MX 2004, que aúna los editores Web Dreamweaver 4 (sencillez) y Dreamweaver UltraDev (amplias funcionalidades). Con esa herramienta se podrá desarrollar cualquier sitio Web personal con características de sitio profesional y utilizar casi todos los recursos de la Web, así como realizar aplicaciones que se ejecuten en servidor y vinculaciones dinámicas de datos; además de contar con un soporte para aplicaciones PHP y utilización de bases MySQL. También cuenta con un amplio soporte para la creación y utilización de CSS (*Cascading Style Sheets*), para lograr un diseño fácil y óptimo.

Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables. Es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML. [20]

El navegador de UML Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen su propia vista de información (vista de Caso de Uso, vista Lógica, vista de Componente y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y funcionalidad del sistema en construcción. [20]

Adobe Photoshop CS2 es el software estándar de edición de imágenes profesional y el líder de la gama de productos de edición de imágenes digitales que aporta más de lo que usted se espera. Las innovadoras herramientas creativas le ayudan a conseguir resultados excepcionales. Una adaptabilidad sin precedentes le permite personalizar Photoshop de acuerdo con su método de trabajo. Además, gracias a unos procesos de edición, tratamiento y gestión de archivos más eficaces podrá trabajar con mayor rapidez. [19]

Se escogió el Adobe Photoshop, como herramienta principal para crear las imágenes del Sistema, ya que se considera la aplicación estándar para el tratamiento digital de imágenes. Las continuas mejoras han hecho de este programa uno de los más profesionales para la edición y retoque fotográfico. Tiene un enfoque dirigido hacia los gráficos para la Web, y posee una total integración con su avanzada herramienta de producción.

2.5. Conclusiones.

En este capítulo se profundizó sobre las tecnologías y metodología actuales a utilizar en el desarrollo del sistema. Además se fundamentó la elección de cuáles herramientas se utilizarán para el desarrollo de la aplicación. Finalmente se llegó a la conclusión de que el sistema se desarrollará sobre el Sistema de Gestión de Contenidos XOOPS, usando el lenguaje de programación PHP, y como gestor de bases de datos se utilizará MySQL.

Capítulo III.

Descripción de la propuesta actual.

3.1. Introducción.

En el presente capítulo se describe la solución propuesta que tiene este trabajo utilizando de algunas de las herramientas que propone la metodología RUP, para ello se describe el modelo de dominio que tiene que ver con el objeto de estudio. Además se realiza una descripción de los requerimientos funcionales y los requerimientos no funcionales, del diagrama de caso de uso y la descripción de cada uno de ellos, así como una descripción general de la solución de sistema que se propone.

3.2. Modelo de Dominio.

Un Modelo del Dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos. [18]

El modelo de dominio tiene como objetivo fundamental visualizar a los usuarios los principales conceptos que se manejan en el dominio a desarrollara a través de la comprensión y descripción de las clases mas importante del sistema.

3.2.1. Glosario de términos:

Curso: Propiedad intelectual, conocimiento y contenidos a ser impartido incluyen texto, audio, vídeo y animación.

Lección: Información detallada sobre las asignaturas y temas del curso que se imparte.

Evaluación: Valoración que se efectúa para determinar que la formación ha cumplido con los objetivos establecidos o ha mejorado el rendimiento y/o conocimientos iniciales del alumno. Proceso de medición del impacto y efectividad de la acción formativa.

Profesor Virtual: Profesional académico que, utilizando los elementos didácticos – pedagógicos y manejando las herramientas de telecomunicación a su alcance, es capaz de conocer a sus asesorado de tal manera que pueda identificar sus necesidades, guiar sus actividades, orientar su aprendizaje fomentando el auto didactismo y estimula la actitud analítica, crítica y constructiva. Es la persona que domina el tema del curso y realiza el diseño instruccional del curso.

Estudiante: Persona que interacciona con un sistema o aplicación de software con el fin de conseguir unos objetivos determinados de aprendizajes.

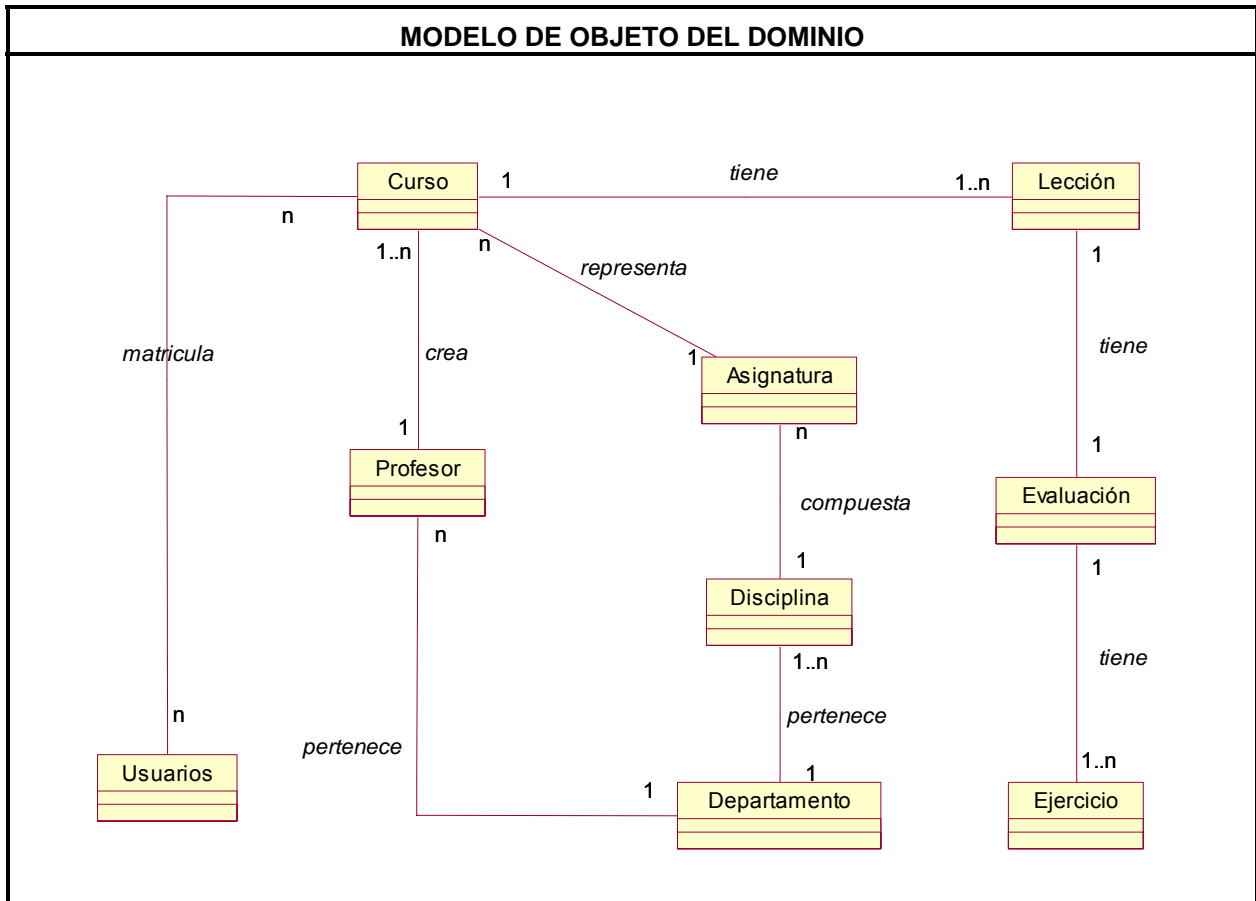
Ejercicios: Pequeñas preguntas escritas que sirven para comprobar el estado de conocimiento de los estudiantes o usuarios del sistema, estas pueden ser: enlazar, seleccionar y marcar.

Asignatura: Materia específica de contenido que se imparte en la educación para la formación de estudiantes.

Disciplina: Conjunto de asignaturas específicas que son una sola pero que se imparte en diferentes niveles.

Departamento: Organización que está compuesta por un conjunto de disciplina relacionada.

3.2.2. Modelo de objeto del dominio



3.3 Requisitos Funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo. [18]

//(Lo que hace el profesor virtual.)

1. Crear ejercicio. (Un crear para cada tipo de Ejercicio)
 - a. Un Ejercicio contiene:
 - i. Enunciado
 - ii. Posibles Respuestas
 - iii. Respuesta correcta

2. Modificar Ejercicio.
3. Crear lección.
 - a. Una lección contiene:
 - i. Tema
 - ii. Título
 - iii. Numero de lección
 - iv. Ejercicios.
4. Modificar lección.
5. Crear curso.
 - a. Un curso contiene:
 - i. Nombre
 - ii. Objetivos
 - iii. Asignatura.
 - iv. Lecciones.
6. Modificar curso.
7. Crea una evaluación.
 - a. Una evaluación contiene:
 - i. Tipo de evaluación.
8. Modificar evaluación.
9. Modificar datos generales.

- // (Lo que solicita el usuario)

10. Autenticar usuario
 - a. El usuario solicita entrar al sistema.
 - i. Entra su nombre de usuario y contraseña de dominio.
 - ii. El sistema lo identifica como profesor virtual o estudiante.
11. Solicitar cursos disponibles.
 - a. Se muestra una lista con los datos de los cursos existentes.
12. Solicitar Matricular curso.
13. Solicitar cursar lecciones.
 - a. Se muestra la lista de lecciones para un curso dado.

14. Solicitar estado.

- a. Se les muestra a los usuarios el estado donde se encuentran, el curso de la última lección que pasaron y el número de esa lección.

15. Escoger curso.

- a. Un usuario escoge matricularse en un curso de la lista de cursos disponibles.

16. Evaluar lección

- a. Un usuario solicita evaluarse en una lección dada.

//(Lo que hace el administrador del sistema)

17. Introducir profesor virtual.

- a. El administrador asigna el rol de profesor virtual a un usuario del dominio UCI.

18. Eliminar profesor virtual.

- a. El administrador le da baja a un profesor virtual.

19. Matricular usuario.

- a. El administrador a pedido de un usuario lo matricula en un curso especificado.
- b. Se le envía un correo de confirmación.

3.4 Requisitos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, como restricciones del entorno o de implementación, rendimiento, etc.

[18]

1. Apariencia o interfaz externa:

- a. Ha de ofrecer un diseño agradable y amistoso.
- b. Debe ofrecer una navegación rápida y sencilla.
- c. Debe tener disponible todas sus opciones.

2. Usabilidad:

- a. El sistema será utilizado por cualquier persona con conocimiento previo de computación y navegadores de Web.
 - b. El sistema podrá ser utilizado desde cualquier lugar donde posea intranet o Internet.
3. Rendimiento:
 - a. Procesamiento de información rápida y eficiente.
 4. Soporte.
 - a. Debe poseer un servidor de base de datos rápido y eficiente.
 - b. Debe interpretarse por cualquier navegador.
 5. Extensibilidad.
 - a. Debe permitir añadir nuevos componentes sin que se afecten el resto de los componentes.
 6. Manejabilidad
 - a. El sistema ha de ser fácilmente manejable y configurable.
 7. Seguridad.
 - a. Autenticar a los usuarios antes que interactúen con el sistema.
 - b. Garantizar que sean los editores los que editen los cursos.
 - c. Protección contra acciones no autorizada en el sistema que puedan afectar la estructura de los datos.
 8. Portabilidad.
 - a. Necesidad que el sistema sea multiplataforma.
 9. Confiabilidad.
 - a. Se garantizan un tratamiento adecuado de las excepciones y la validación de las entradas del usuario.

3.5 Descripción del sistema propuesta.

Par dar cumplimiento a los objetivos trazado desde un inicio en este trabajo y llevando en cuenta todos los requisito requerido por el sistema, el sistema está compuesto por dos módulos principales, uno el módulo de administración encargado de administrar el sistema y un módulo de edición, que permite mostrar a los usuarios diferentes cursos que son editados por los profesores virtuales.

El módulo administración del sistema permite que el editor pueda editar sus cursos y manipularlos, también permite que los distintos usuarios matriculen convirtiéndose en parte del sistema y tenga la posibilidad ver todos los cursos disponible y escoger el de su preferencia para cursarlo.

El sistema se analizó y se consideró que estaba compuesto por tres roles, un rol de administrador que es el cargado de administrar el sistema y adiciona al sistema los clientes dándole su privilegios según su categoría, un rol de profesor virtual que es encargado de de la edición de el curso y su contenido, y un rol de usuario que es el que solicita el curso y su contenido.

El modulo edición será manipulado nada más por los profesores asignados para impartir los cursos a los estudiantes y ellos son los encargados editar y mantener una buena información dentro de los curso que van hacer presentado. El módulo administración solo será utilizado por personas escogida que sean responsables de administrar el sistema y de llevar un control sobre los editores y estudiantes que acceden al sistema.

3.5.1 Definición de los actores del sistema a automatizar.

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interaccionan con estos. Los actores representan terceros fuera del sistema que colaboran con el mismo. Una vez que hemos identificado los actores del sistema, tenemos identificado el entorno externo del sistema. [18]

Actores	Descripción	Justificación
----------------	--------------------	----------------------

Usuario	El usuario representa a una persona que accede al sistema para pasar uno de los cursos disponibles, estudiar sus lecciones, realizar las pruebas y ver su estado dentro de un determinado curso	R10-R16
Administrador	El administrador representa a una persona responsable del sistema es la encargada de permitir que los usuarios forme parte del sistema y de dar permiso a los editores para que publiquen los cursos.	R17, R18
Profesor Virtual	El profesor virtual representa a una persona encargada de establecer los cursos en el sistema y chequear el estado de los usuarios.	R1-R9

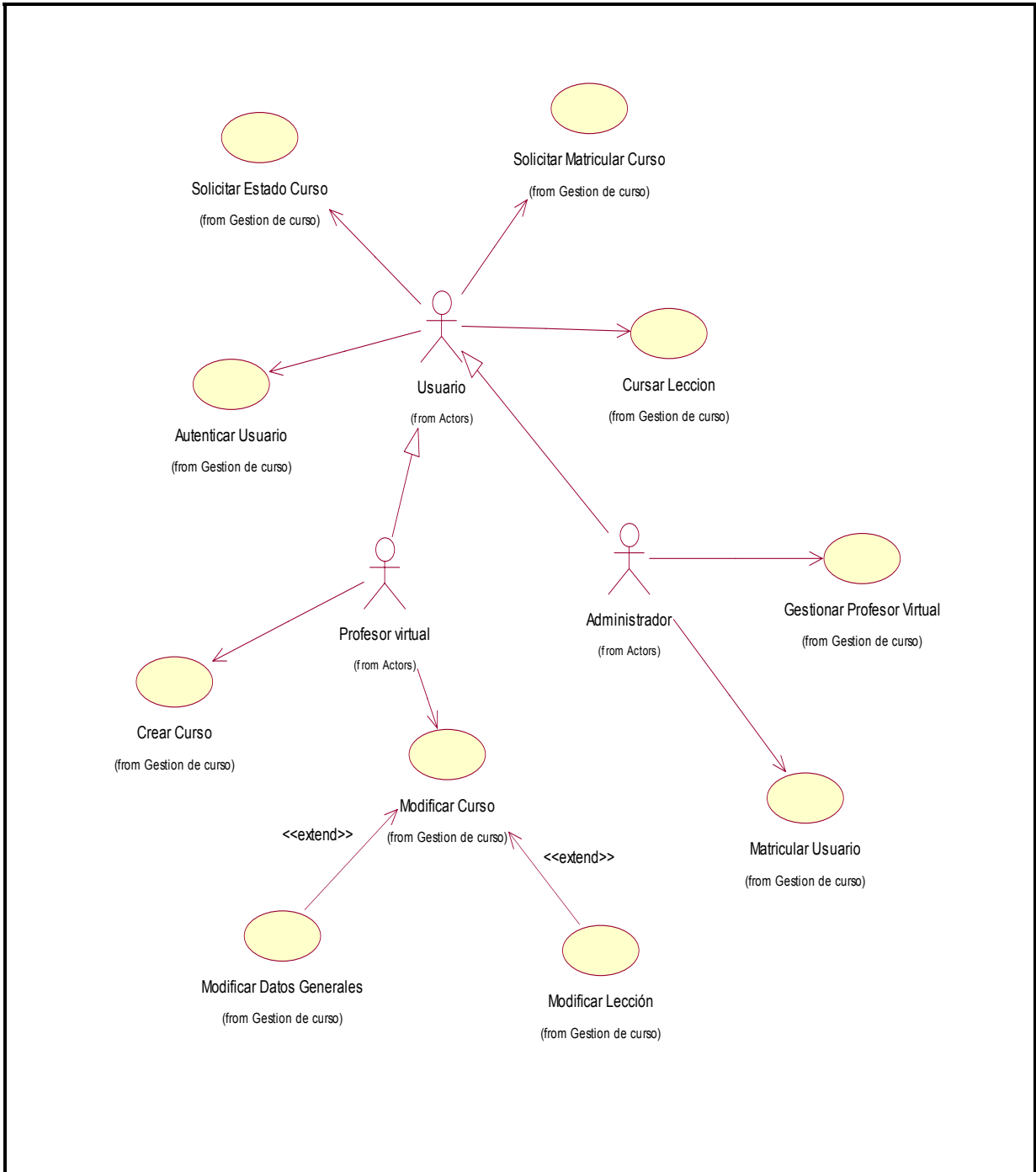
3.5.2 Definición de los casos de uso del sistema.

El modelo de casos de uso (use case model) documenta el comportamiento del sistema desde el punto de vista del usuario, permitiendo representar las funciones que se desean en el sistema (use cases), el entorno del sistema (actores), y las relaciones entre ellos. Aunque la parte más visible de dicho modelo son los diagramas de casos de uso, suele ir acompañado de una especificación textual de cada uno de los casos de uso.

Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. El modelado de casos de uso es la técnica más efectiva y a la vez la más simple para modelar los requisitos del sistema desde la perspectiva del usuario. Los casos de uso se utilizan para modelar cómo un sistema o negocio funciona actualmente, o cómo los usuarios desean que funcione. No es realmente una aproximación a la orientación a objetos; es realmente una forma de modelar procesos. Es, sin embargo, una manera muy buena de dirigirse hacia el análisis de sistemas orientado a objetos. Los casos de uso son generalmente el punto de partida del análisis orientado a objetos con UML. [23]

Los principales objetivos del modelo de casos de uso son: permitir la comunicación entre los desarrolladores y los clientes o usuarios durante la captura de requisitos y planificar las iteraciones necesarias en el proceso software y ser la base para la validación del sistema.

Los casos de uso no son parte del diseño, sino parte del análisis. De forma que al ser parte del análisis nos ayudan a describir qué es lo que el sistema debe hacer. Los Casos de uso son lo que hace el sistema desde el punto de vista del usuario, es decir, describen un uso del sistema y cómo este interactúa con el usuario.



Nombre del caso de uso		Modificar Lección <<extend>>
Actores	Caso de Uso: Modificar Curso (Caso de Uso Base).	
Propósito	Permitir realizar transformaciones, cambios en el contenido de una lección de forma rápida y eficiente.	
Resumen:	Estando el curso seleccionado, el profesor virtual solicita hacer transformaciones en el contenido de una lección determinada, entra los datos, el sistema chequea que estos datos sean los correctos y estas transformaciones son registradas en la base de datos.	
Requisitos	Asociados: R4.	
	Caso de Usos Asociados: Modificar Curso (Caso de Uso Base).	
Precondiciones	Curso seleccionado al que pertenece la lección seleccionada. Existencia en la base de datos de la lección a modificar.	
Poscondiciones	Lección actualizada en la base de datos.	

Nombre del caso de uso		Crear Curso.
Actores	Profesor Virtual	
Propósito	Permitir la edición y montaje de un curso, perteneciente a una disciplina, de forma rápida y eficiente.	
Resumen:	Un profesor virtual selecciona la disciplina y la asignatura a la cual pertenece el curso. Solicita editar y montar un curso en el sistema, realiza la entrada de datos necesarios, el sistema chequea los datos generales y se almacena la información en la base de datos.	
Referencias	Requisitos asociados: R1, R3, R5.	
Precondiciones	Existencia de al menos una disciplina y una asignatura por disciplina a la cual pertenece el curso seleccionado.	
Poscondiciones	Curso almacenado en la base de datos.	

Nombre del caso de uso		Modificar Curso
Actores	Profesor Virtual	
Propósito	Realizar transformaciones a los datos de un Curso dada la selección del actor.	
Resumen:	Un profesor solicita realizar modificación en un curso. Selecciona si estas transformaciones van a ser a los datos generales o a las lecciones en cuyo caso son ejecutados los	

	casos de usos extendidos. Realiza las modificaciones y los cambios son actualizados en la base de datos.
Referencias	Requisitos asociados: R4, R9.
	Casos de uso asociados: Modificar Datos Generales. <<extend>> Modificar Lección. <<extend>>
Precondiciones	Listado de los posibles cursos a modificar mostrados. Curso existente en la base de datos.
Poscondiciones	Curso modificado actualizado en la base de datos.

Nombre del caso de uso		Modificar datos generales <<extend>>
Actores	Caso de Uso: Modificar Curso (Caso de Uso Base).	
Propósito	Permitir realizar transformaciones, cambios en los datos generales de forma rápida y eficiente	
Resumen:	Estando el curso seleccionado, el profesor virtual solicita hacer transformaciones en los datos generales del curso, entra los datos, el sistema chequea que estos datos sean los correctos y estas transformaciones son registradas en la base de datos.	
Referencias	R9	
	Caso de Usos Asociados: Modificar Curso (Caso de Uso Base).	
Precondiciones	Curso al que pertenecen los datos generales asociados seleccionado. Existencia en la base de datos del curso a modificar.	
Poscondiciones	Datos generales del curso seleccionado actualizados en la base de datos.	

Nombre del caso de uso		Autenticar usuario
Actores	Usuario	
Propósito	Permitirle al usuario autenticarse en el sistema para poder hacer uso de los servicios.	
Resumen:	Un usuario entra sus datos de identificación. El sistema verifica la existencia en el dominio y qué permisos tiene en el sistema. Si los datos introducidos son correctos el usuario tiene acceso al módulo.	
Referencias	R10	
Precondiciones	Debe existir un usuario en el dominio con esos datos de identificación.	

Poscondiciones	Usuarios Autenticados y permisos de trabajos asignados si los datos son correctos. El usuario tiene acceso al módulo si los datos son correctos.
----------------	---

Nombre del caso de uso	Solicitar matricular cursos
Actores	Usuario
Propósito	El usuario solicita la matricula un curso dada su selección.
Resumen:	Un usuario entra al sistema después de ser autenticado y solicita ver un listado de los cursos disponibles. Selecciona un curso a matricular y el sistema gestiona la petición del usuario.
Referencias	R11, R12, R15.
Precondiciones	Debe existir al menos un curso en la base de datos.
Poscondiciones	Solicitud del usuario realizada. Gestión de la matrícula del curso seleccionado iniciada.

Nombre del caso de uso	Cursar lección
Actores	Usuario
Propósito	Permitirle al usuario cursar la lección seleccionada por este y realizar todas las acciones contenidas en la lección.
Resumen:	Un usuario selecciona uno de los cursos matriculados, escoge la lección a cursar, el sistema verifica si ya el usuario cursó la lección y actualiza las necesidades de evaluación. El usuario realiza todas las operaciones contenidas en la lección.
Referencias	R13, R16.
Precondiciones	Usuario matriculado en el curso al que pertenece la lección. Debe existir al menos una lección del curso especificado en la base de datos.
Poscondiciones	Lección cursada. Base de datos actualizada.
Nombre del caso de uso	Solicitar estado curso
Actores	Usuario
Propósito	Dar a conocer a los usuarios el estado en que se encuentra dentro de un curso.
Resumen:	Un usuario solicita ver su estado dentro de un curso específico. El sistema le muestra toda la información de su desempeño hasta donde ha cursado.
Referencias	R14

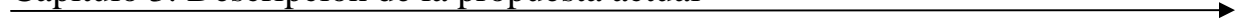
Precondiciones	El usuario tiene que haber vencido al menos una lección del curso seleccionado.
Poscondiciones	Visualizado el estado del usuario en el curso.

Nombre del caso de uso	Gestionar profesor virtual
Actores	Administrador
Propósito	Crear y eliminar los profesores virtuales en el sistema.
Resumen:	El administrador del sistema selecciona la acción a realizar (Crear o Eliminar). Realiza la selección específica y el sistema actualiza la base de datos.
Referencias	R17,18
Precondiciones	Usuarios profesores existentes en el dominio.
Poscondiciones	Información de los profesores virtuales actualizada en la Base de Datos.

Nombre del caso de uso	Matricular usuario
Actores	Administrador
Propósito	Aprobar la solicitud de matrícula de un usuario.
Resumen:	El administrador del sistema aprueba la solicitud de matrículas de los usuarios seleccionando a estos en el listado de solicitantes a matrículas. El sistema actualiza la base de datos y notifica a los usuarios matriculados.
Referencias	R19
Precondiciones	Al menos debe existir una solicitud de matrícula de un usuario en el curso seleccionado.
Poscondiciones	Usuario matriculado. Notificación de matrícula al usuario enviada. Base de datos actualizada.

3.6. Conclusiones.

En este capítulo se empezó a describir la solución propuesta de trabajo partiendo a partir del análisis de proceso de dominio, donde se definieron un listado de requisitos y un listado de casos de usos que son representado con un diagrama y expandido su desarrollo



Capítulo IV.

Construcción de la propuesta de solución.

4.1 Introducción.

En el presente capítulo se realiza una descripción de la construcción de la solución propuesta donde este trabajo hace uso de un Sistema de Gestión de Contenido, en este caso XOOPS. Para realizar trabajos con XOOPS se debe seguir una metodología determinada debido a como está construida la arquitectura de este Sistema de Gestión de Contenido. Por lo general los CMS tienen como funcionalidad la de trabajar como una plataforma dinámica, tienen consigo dentro de su arquitectura una propiedad que les permite adicionar diferentes módulos con distintas funcionalidades. En este capítulo se realiza la modelación de módulos que dan solución a la propuesta de trabajo mediante los diagramas de diseño y diagrama de clase persistente.

4.2. Diagrama de clases.

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc.). Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones.

[23]

El Diagrama de Clase es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones.

Los paquetes son:

Paquete 1: Perfil de usuario.

Sub-Paquete 1.1: Cursar Curso

Sub-Paquete 1.2: Solicitar Estado Curso.

Sub-Paquete 1.3: Solicitar Matricular Curso.

Paquete 2: Perfil de Profesor.

Sub-Paquete 2.1: Modificar Curso.

Sub-Paquete 2.2: Crear Curso.

Paquete 3: Perfil de Administración.

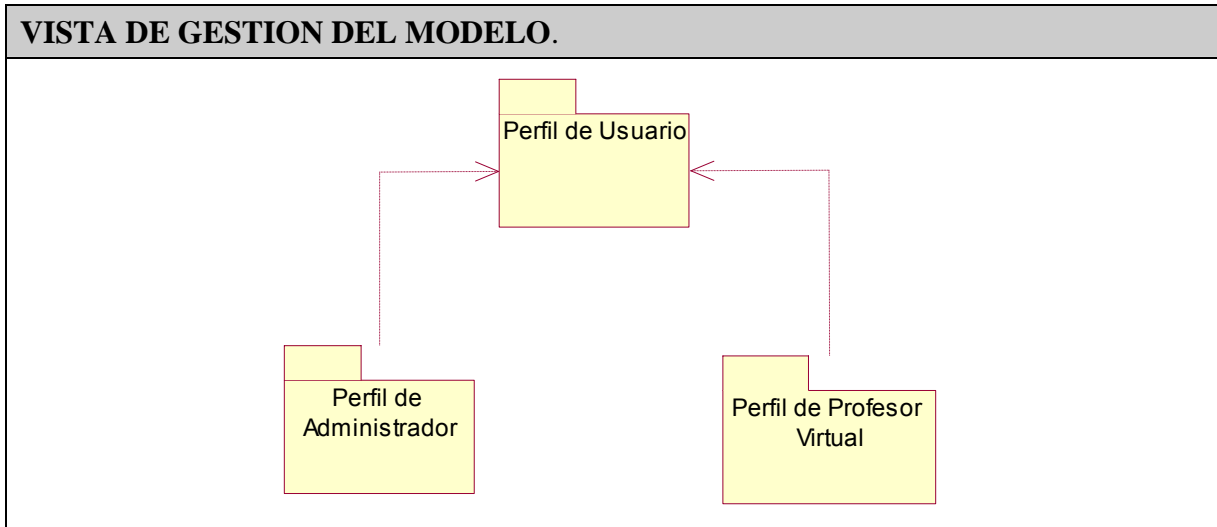
Sub-Paquete 3.1: Matricular Usuario.

Sub-Paquete 3.2: Gestionar Profesores Virtuales.

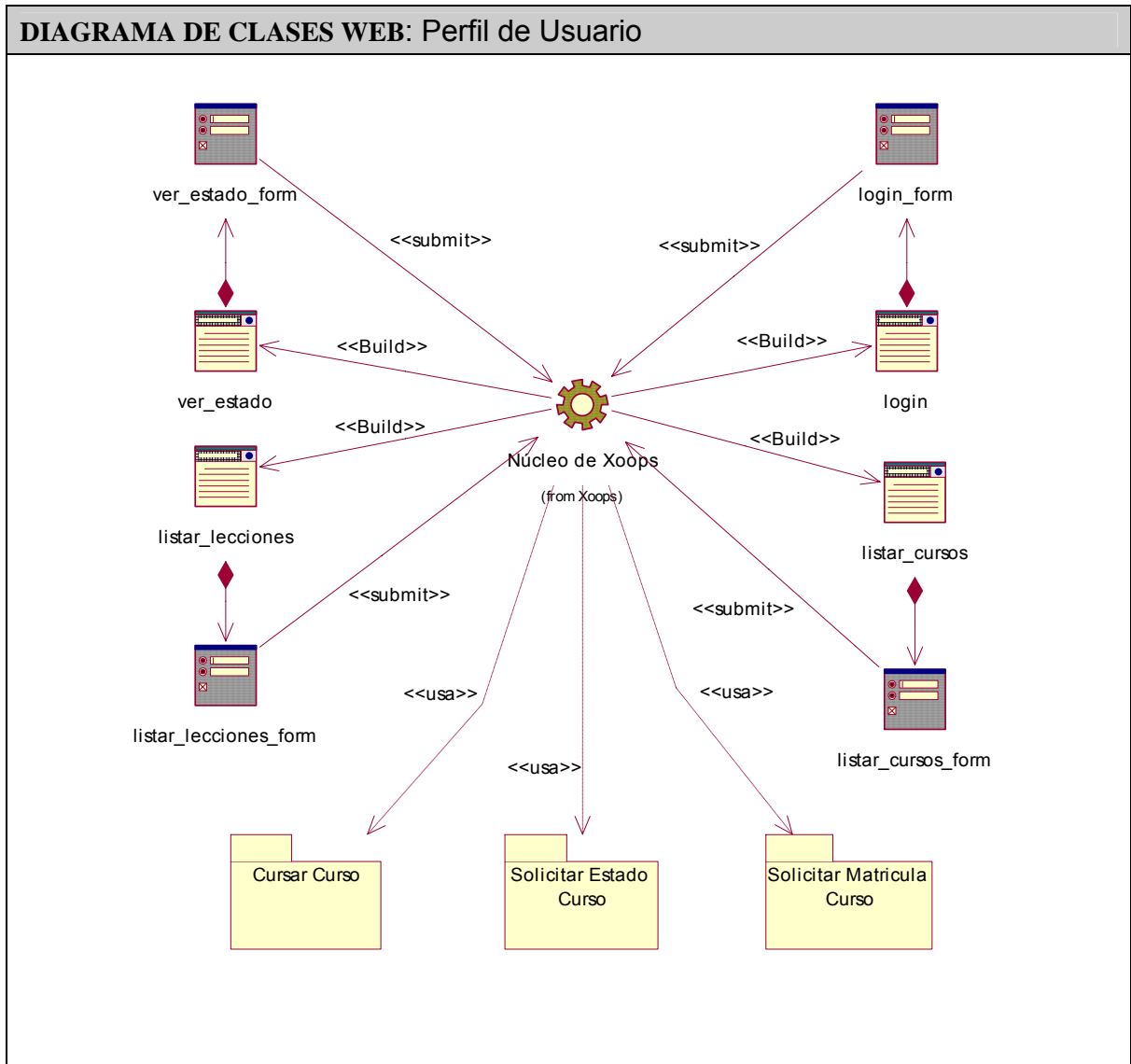
Para una mejor comprensión de los diagramas y del modelo se especificara a continuación la nomenclatura utilizada en las clases.

Clases	Significado
DAL	Data Access Layer, representa una clase utilizada para el acceso a datos
BLL	Business Logic Layer, representa una clase que implementa reglas de negocio.
Common	Comunes. Todas las clases comunes.
Form	Representa un formulario WEB.
Admin.	Representa las clases definidas con fines de administración.
Blocas	Representa un bloque del modulo.

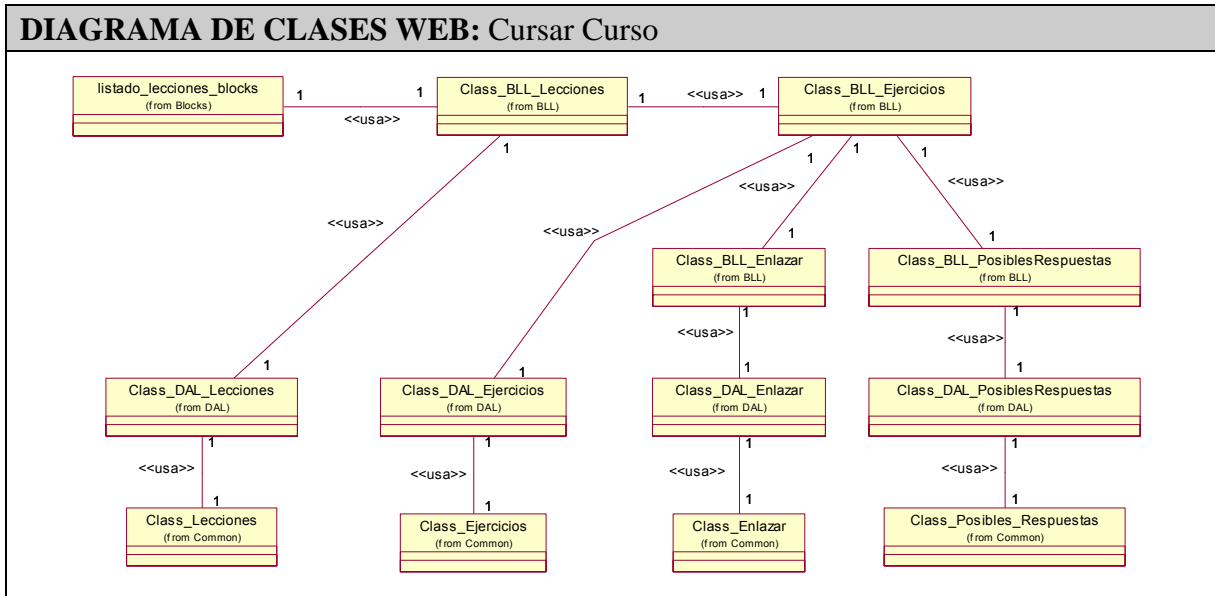
Vista de gestión del modelo



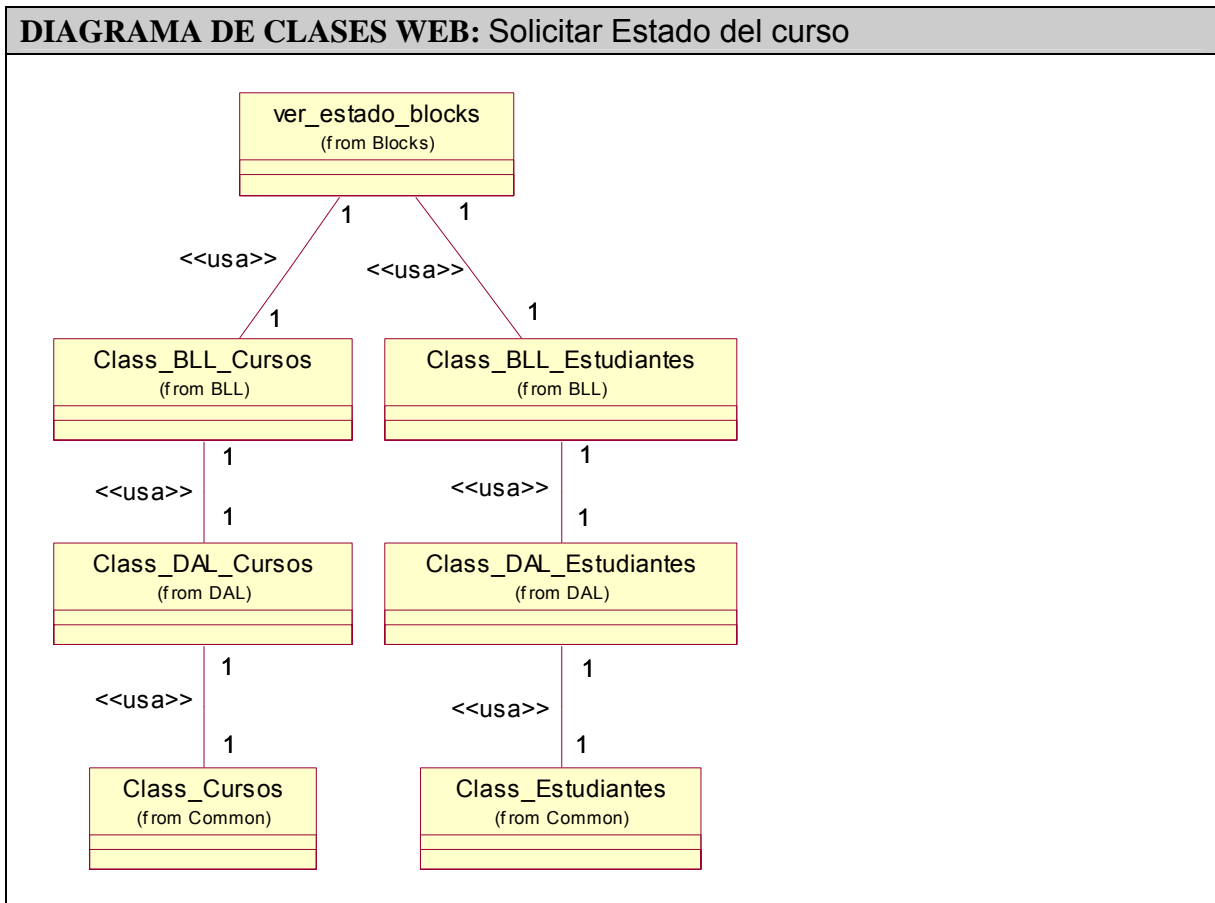
4.2.1. Paquete 1: Perfil de Usuario.



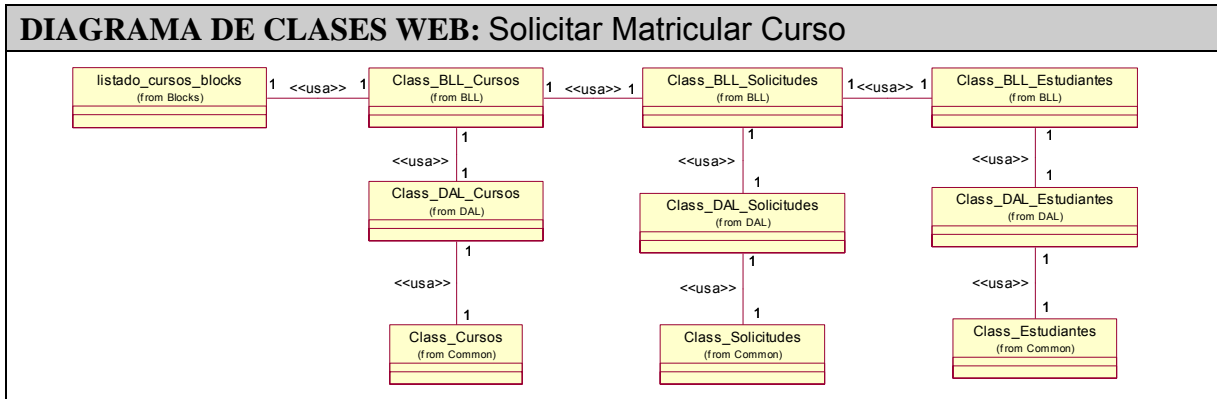
4.2.1.1 Sub-Paquete1.1: Cursar Curso.



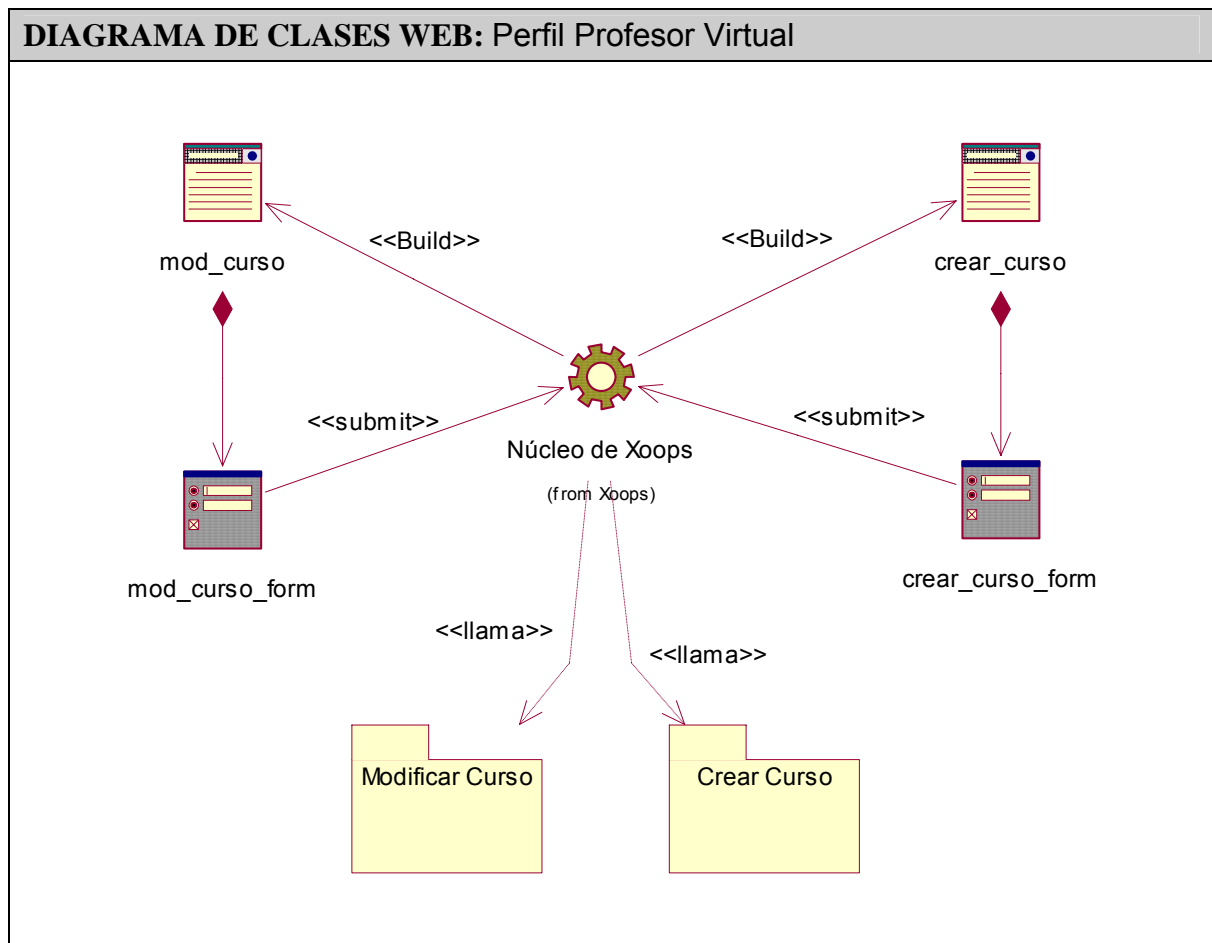
4.2.1.2 Sub-Paquete1.2: Solicitar Estado del Curso.



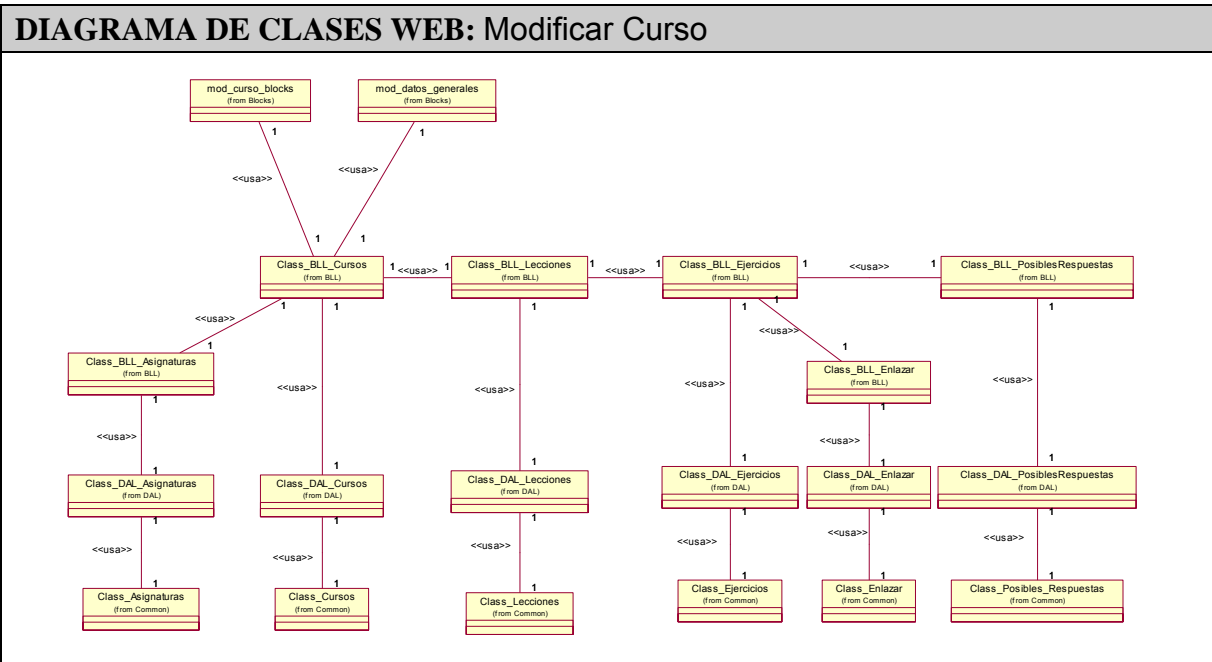
4.2.1.3 Sub-Paquete 1.3: Solicitar Matricular Curso



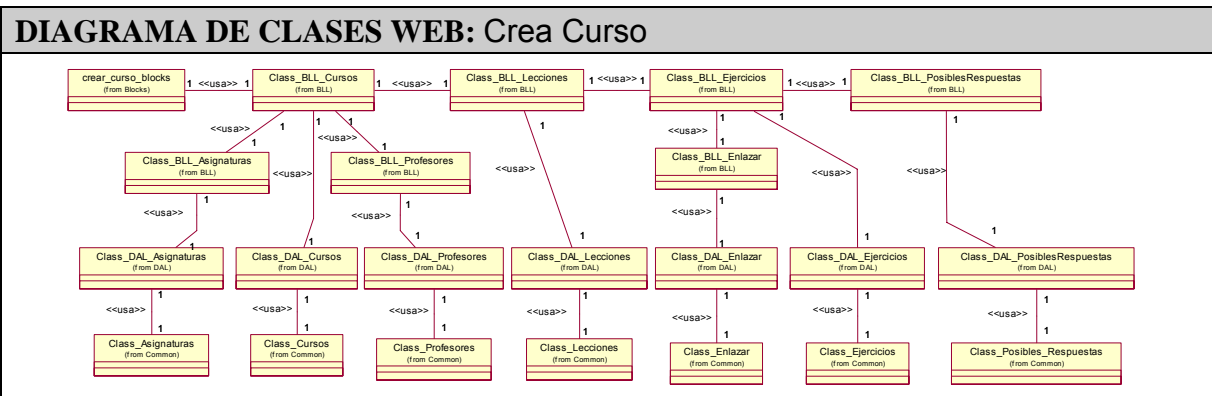
4.2.2. Paquete 2: Perfil Profesor Virtual.



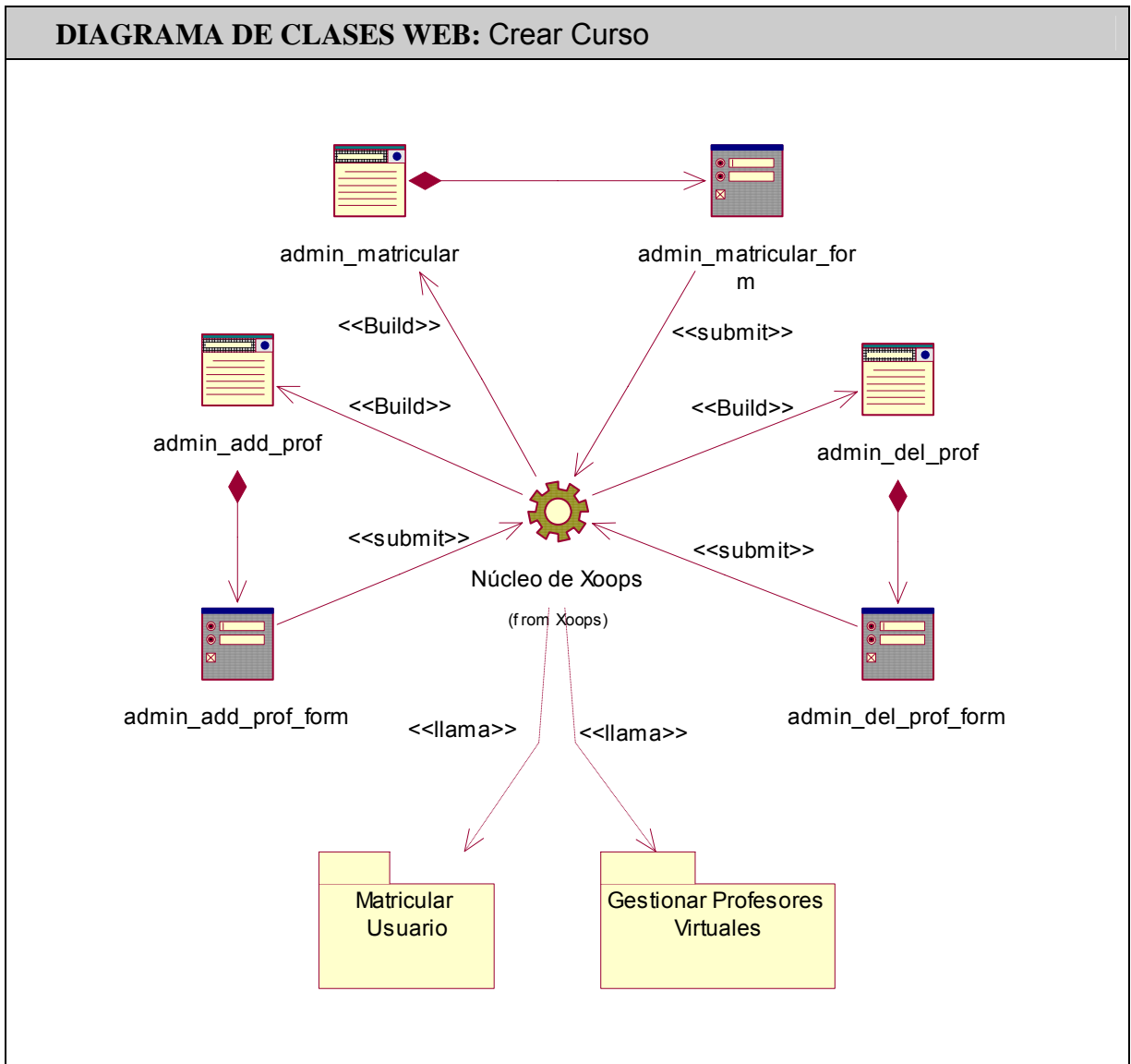
4.2.2.1 Sub-Paquete2.1: Modificar curso



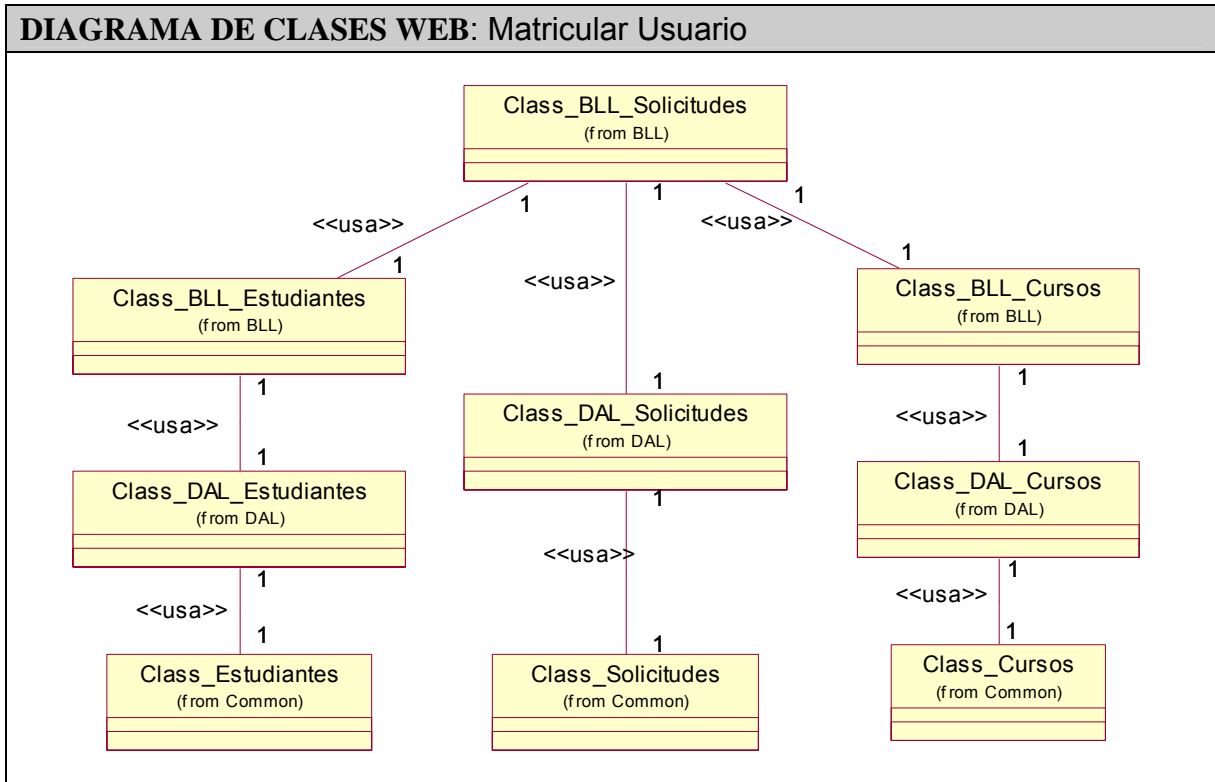
4.2.2.2 Sub-Paquete2.1: Crea Curso.



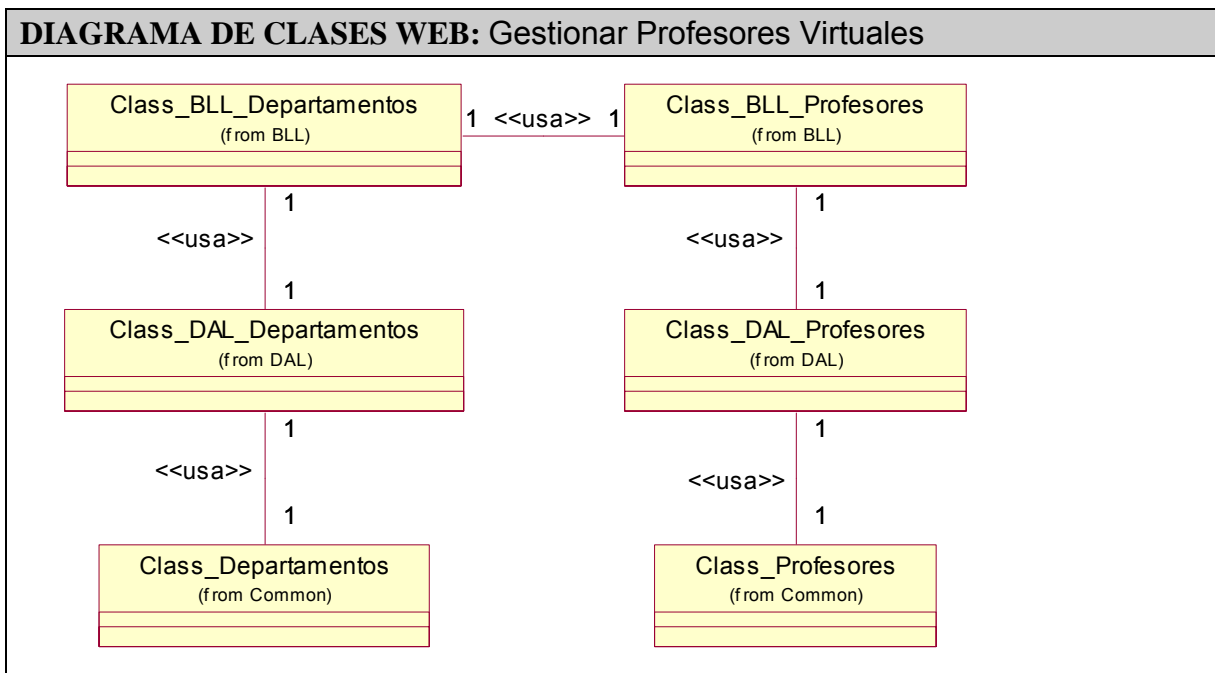
4.2.3. Paquete 3 Perfil Administración.



4.2.3.1 Sub-Paquete3.1: Matricular Usuario



4.2.3.1 Sub-Paquete3.2: Gestionar Profesores virtuales.



4.3. Modelo de datos.

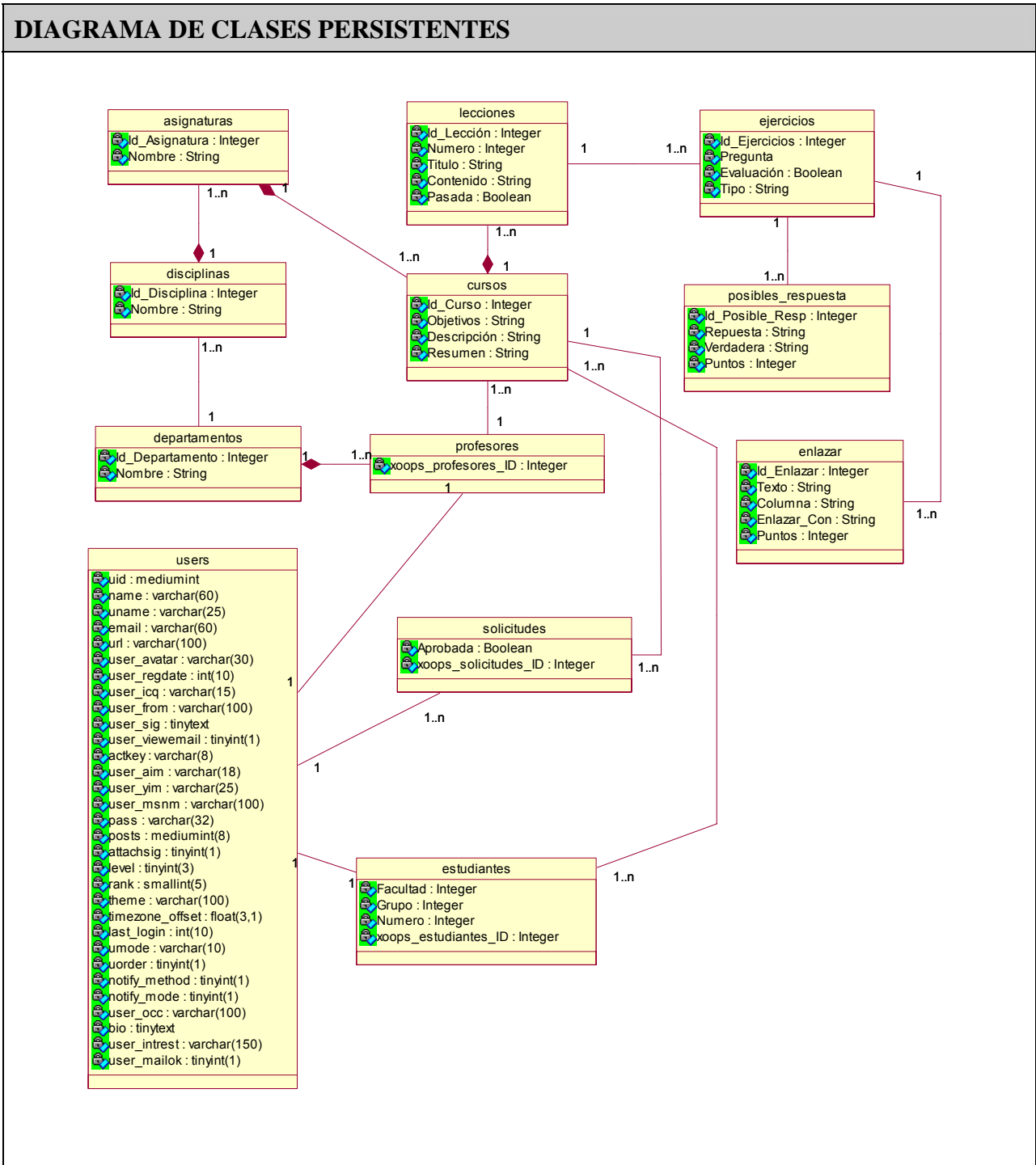
Un modelo de datos es un sistema de signos con funciones semióticas que tiene una capacidad semántica que permite la representación del conocimiento sobre el dominio de la aplicación y facilita la comunicación humano-humano y humano-computador. Los componentes de un modelo de datos son:

- Las estructuras básicas o tipos de datos empleados para modelar o representar el conocimiento.
- Las operaciones que permiten la creación, manipulación, consulta y administración de los datos, que representan el comportamiento dinámico de las estructuras básicas.
- Las reglas de formación que determina como se usa el modelo y las reglas de integridad, que aseguran la consistencia de los datos. [24]

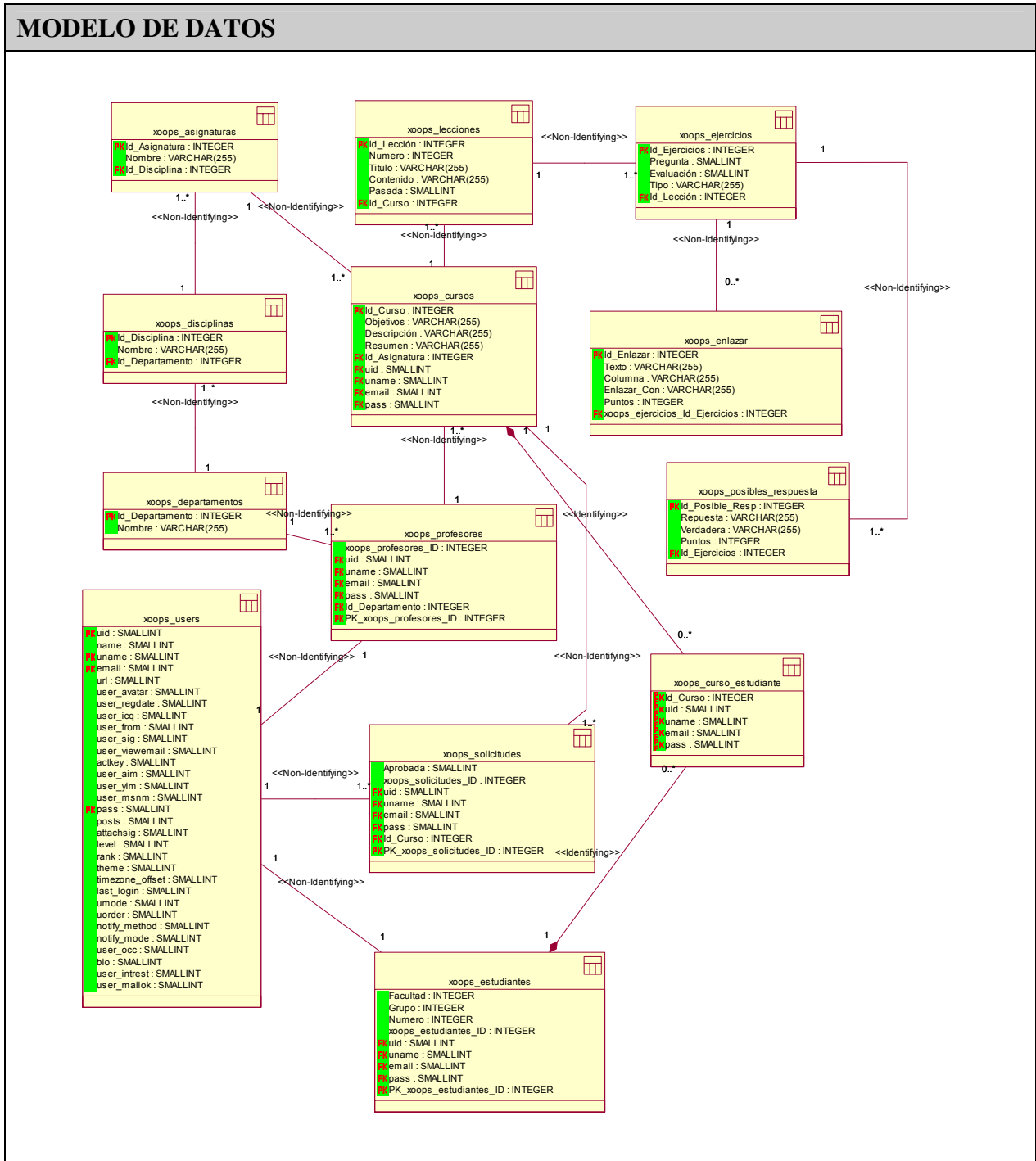
La parte de UML que esta relacionada con los datos se denomina Diagrama de Clase. A partir de este diagrama y los requerimientos del sistema se puede lograr el modelo conceptual de un Modelo de Datos, aplicando las transformaciones que más se adapten al sistema que esta en estudio.

El Diagrama de Clase presenta un mecanismo de implementación neutral para modelar los aspectos de almacenado de datos del sistema. Las clases persistentes, sus atributos, y sus relaciones pueden ser implementados. Aun así, en el entorno de desarrollo actual, la base de datos relacional es el método más usado para el almacenamiento de datos. El diagrama de clase de UML se puede usar para modelar algunos aspectos del diseño de bases de datos relacionales.

4.3.1. Diagrama de clases persistentes.



4.3.2. Diagrama de Datos.



4.4. Principios del diseño, Manejo de Errores y Estándares de codificación.

4.4.1 Principios de Diseño.

El diseño debe basarse en el usuario, para ello al Xoops se le pueden aplicar skins (pieles) para obtener una mejor vista. Estas pieles se controlan con los llamados Themes (temas), estos temas contienen código HTML. Esto le permite al diseñador del sitio cambiar la piel sin que se sepa nada de programación.

Para ello, este sistema utiliza ciertos principios generales que garantizan la usabilidad en los diseños para aplicaciones Web.

1. Principio de uso equiparable: donde las características de privacidad, garantía y seguridad estén igualmente disponibles para todos los usuarios, y que el diseño sea atractivo para todos los usuarios.
2. Principio de la flexibilidad: donde se ofrezcan posibilidades de elección en los métodos de uso, que facilite al usuario la exactitud y precisión, y se adapte al paso o ritmo del usuario.
3. Principio de la Información perceptible: donde se usen diferentes modos para presentar de manera redundante la información esencial (gráfica y verbal), se proporcione contraste suficiente entre la información esencial y sus alrededores, se amplíe la legibilidad de la información esencial, y que diferencie los elementos en formas que puedan ser descritas (por ejemplo, para las funciones de catalogación).
4. Principio de tolerancia al error: donde se dispongan los elementos para minimizar los riesgos y errores, por ejemplo utilizando elementos comunes; y los elementos peligrosos eliminados, aislados o tapados, que se proporcionen advertencias sobre peligros y errores. Hay que posibilitar el descubrimiento interactivo y el aprendizaje ensayo-error, y posibilitar la reversibilidad y la recuperabilidad de las acciones.

5. Principio de esfuerzo de acceso y uso: que minimicen las acciones repetitivas, y que proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie.

4.4.2 Manejo de Errores.

El Xoops tiene un buen manejo de errores. Para ello se apoya en la clase XoopsErrorHandler que se encuentra en xoops/class/errorhandler.php y tiene las funciones XoopsErrorHandler_HandleError donde te dice el tipo de error, el mensaje de error, el fichero en el que ocurrió el error, así como la línea en que ocurre el error. Y XoopsErrorHandler_Shutdown que se ejecuta al salir. Una función disponible para el usuario es xoops_error, a la cual se le pasa el mensaje de error y el título del mensaje. Muy importante para que el usuario notifique de algún error ocurrido.

También puedes enviar mensajes para la confirmación de borrado. Esto es muy importante porque así evitas que se elimine algún registro de la tabla por accidente. Para ello se utilizan la función de confirmación xoops_confirm.

Xoops controla de forma excelente el control de los usuarios a áreas no autorizadas. En la Figura 1 muestra un ejemplo de esto:

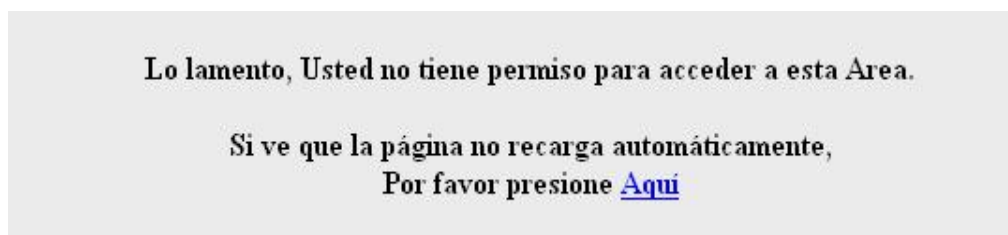


Figura 1

4.4.3 Estándares de Codificación.

"Mientras menos se entienda mi código mejor es". Esta es una filosofía que se seguía en la programación. Ahora todo es diferente pues si tenemos un código claro y

comprensible, esto facilita la corrección de algún problema en módulo. Por ello tuvimos en cuenta las siguientes normas:

Bloques de instrucciones: el corchete de inicio al final de la instrucción y el corchete de fin al inicio alineado con la línea de instrucción. El código que va dentro a una profundidad de tres espacios.

Comparaciones seguras: siempre escribir primero los números en las comparaciones de igualdad. Esto garantiza que no asigne valores a variables.

Nombres de las clases: comienza con la palabra **Class** y el nombre de la clase, y están separados por _ (guión inferior) para mayor legibilidad.

Atributos de las clases y variables: siempre se escribieron con minúsculas.

Parámetros y argumentos: se puso el nombre del parámetro en correspondencia con lo que se que ría pasar y siempre con una **p** delante de la palabra.

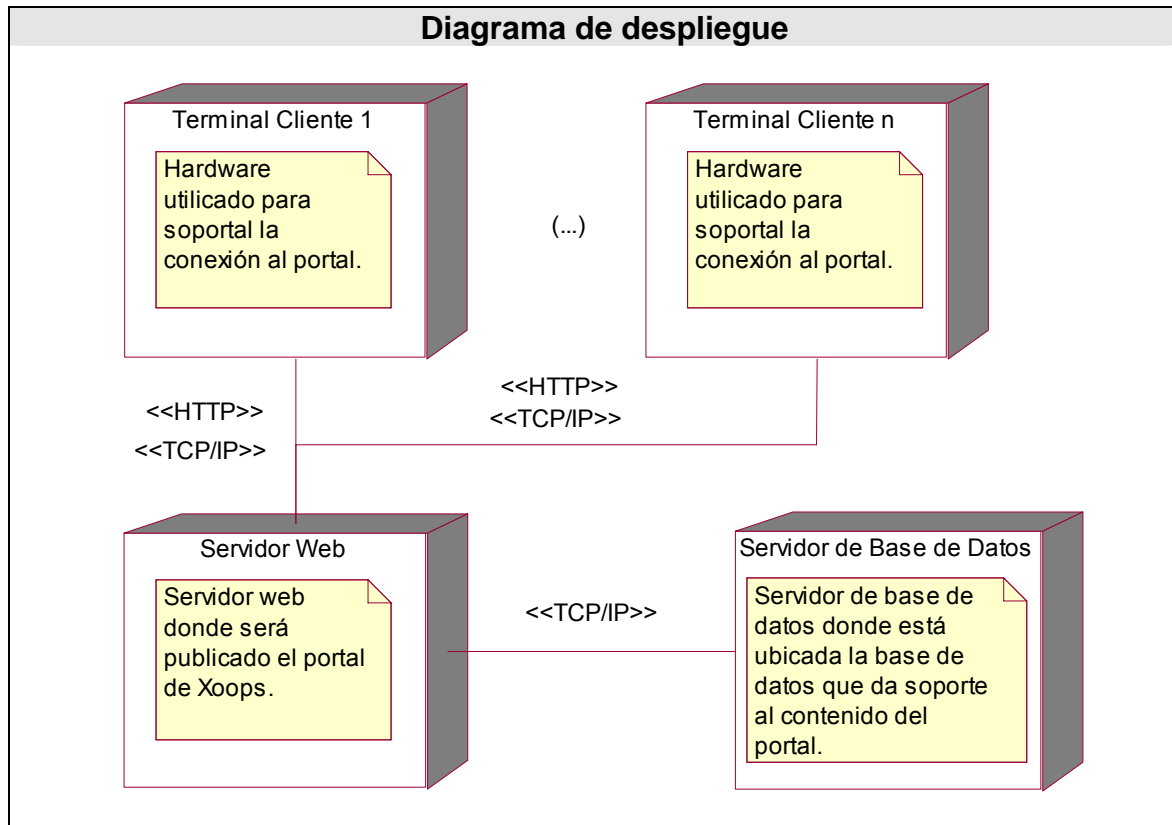
Métodos y funciones: la primera letra es mayúscula y en caso que contenga dos palabras se separan por _ (guión inferior).

4.5. Diagrama de despliegue.

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes). [25]

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad

de computación de algún tipo, desde un sensor a un mainframe. Las instancias de componentes software pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz). [25]



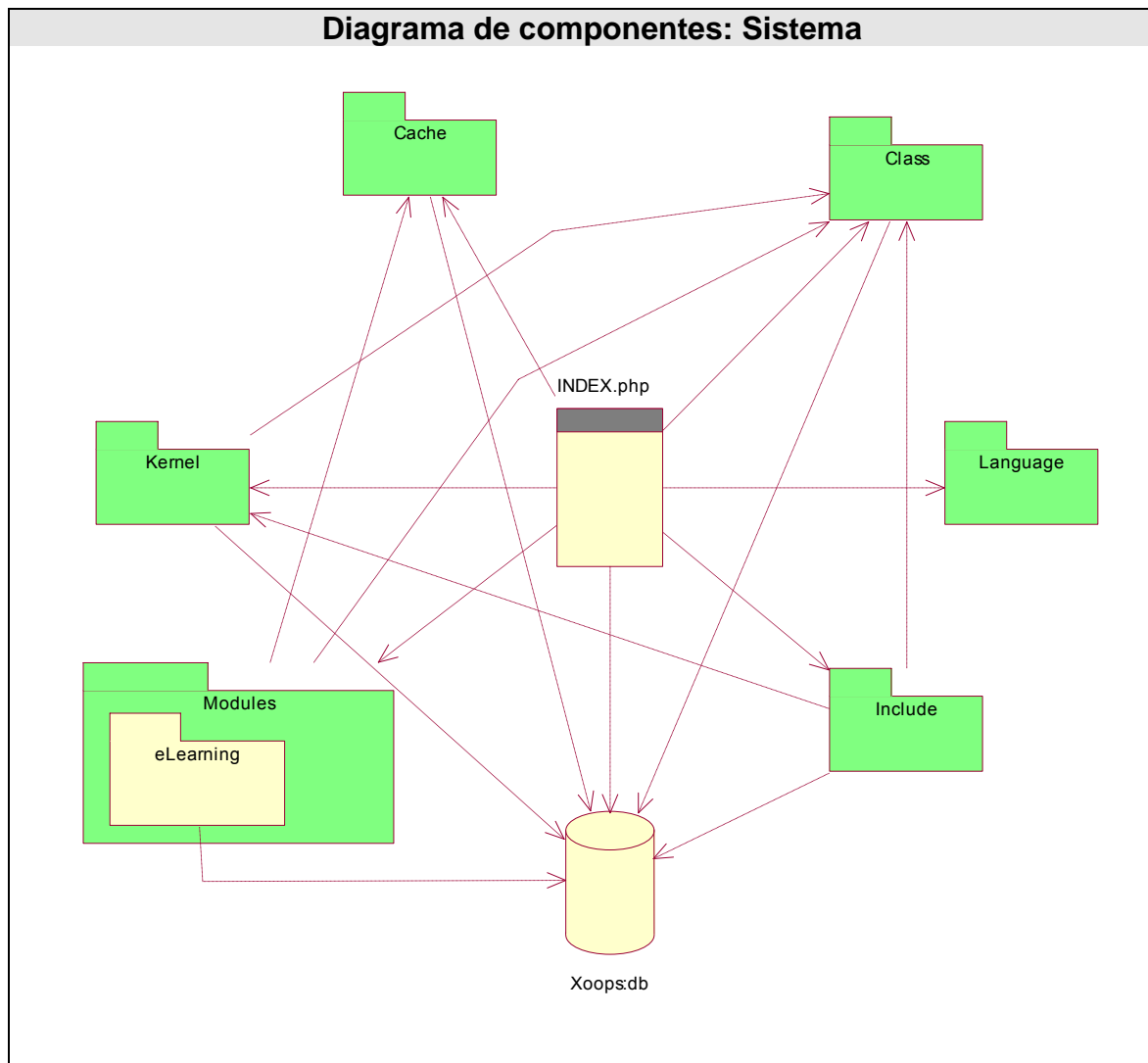
4.6. Diagrama de componentes.

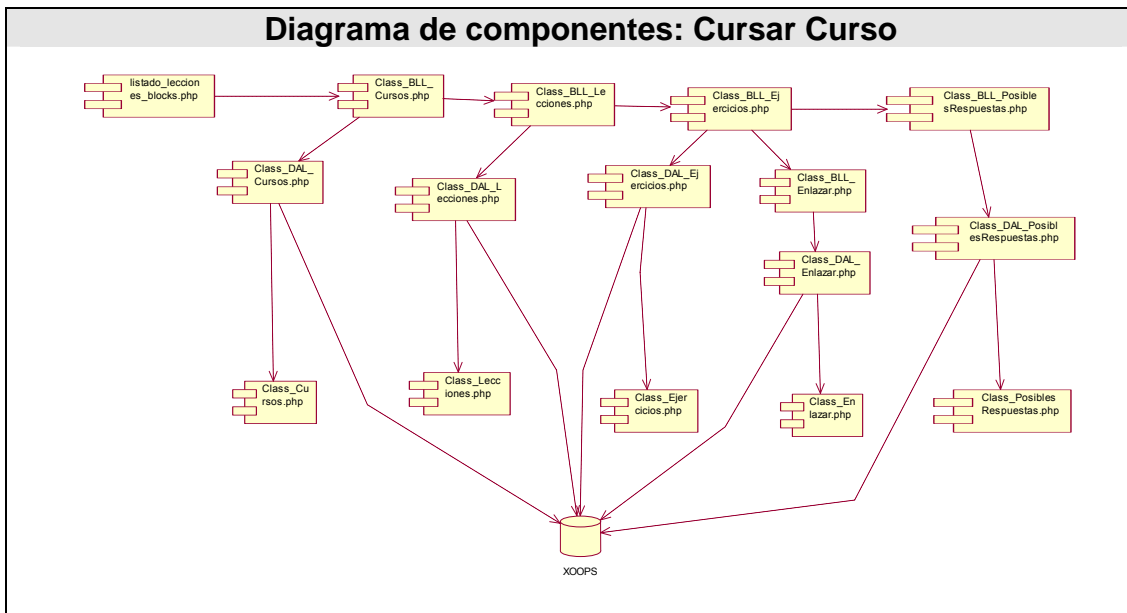
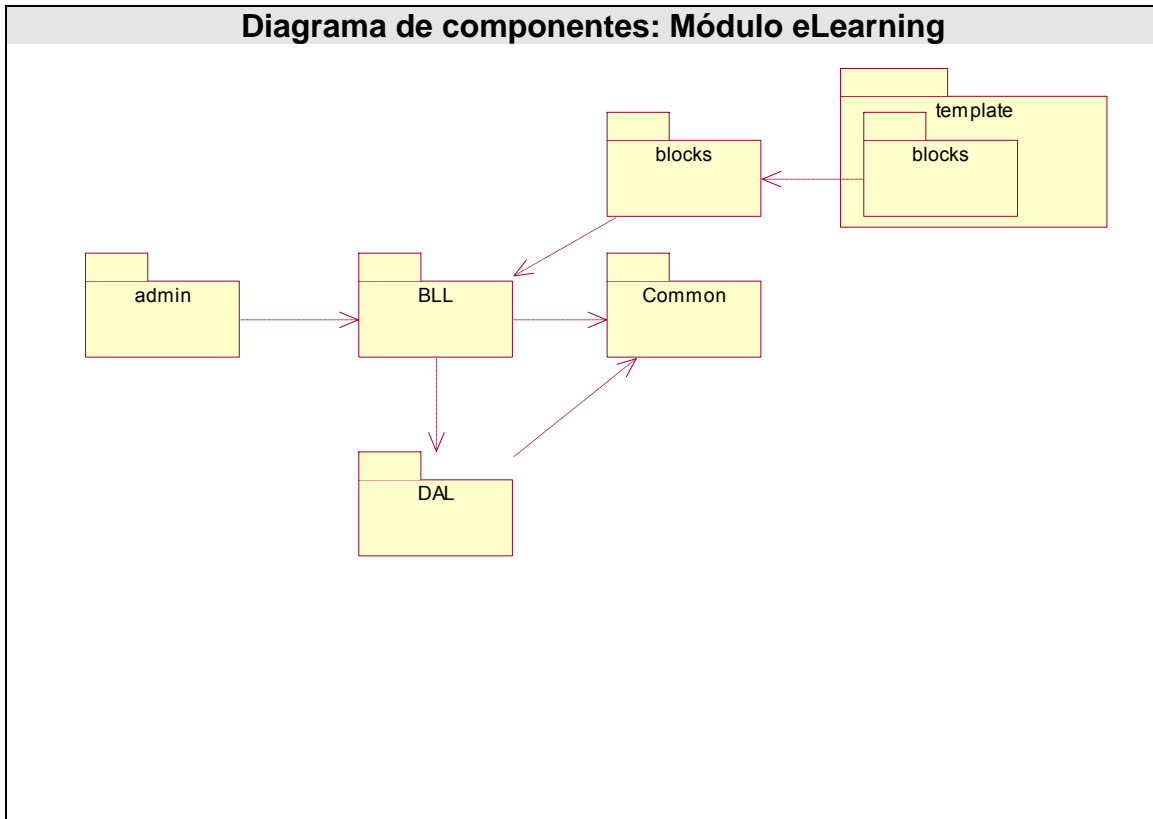
Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo. [25]

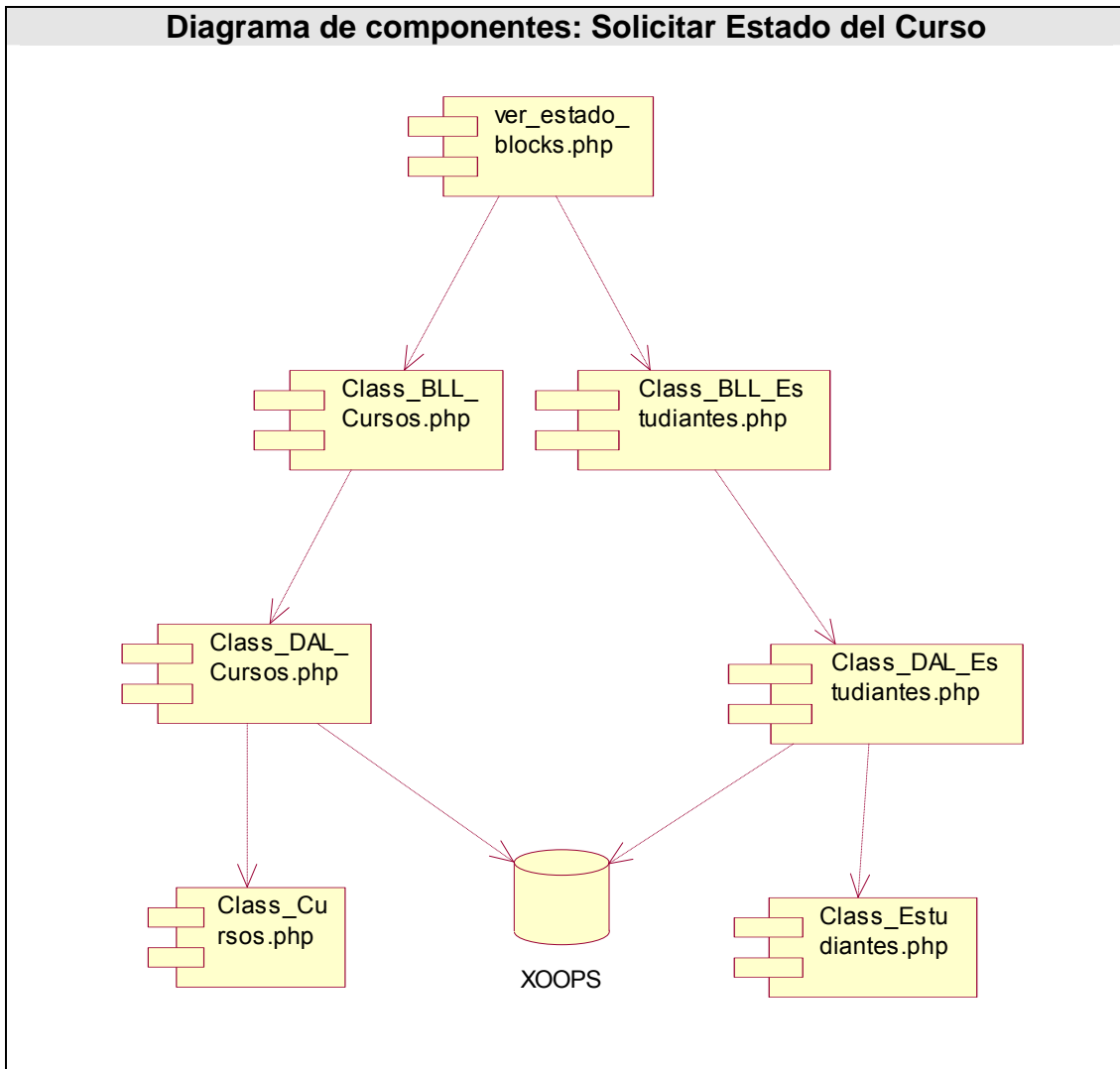
Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en

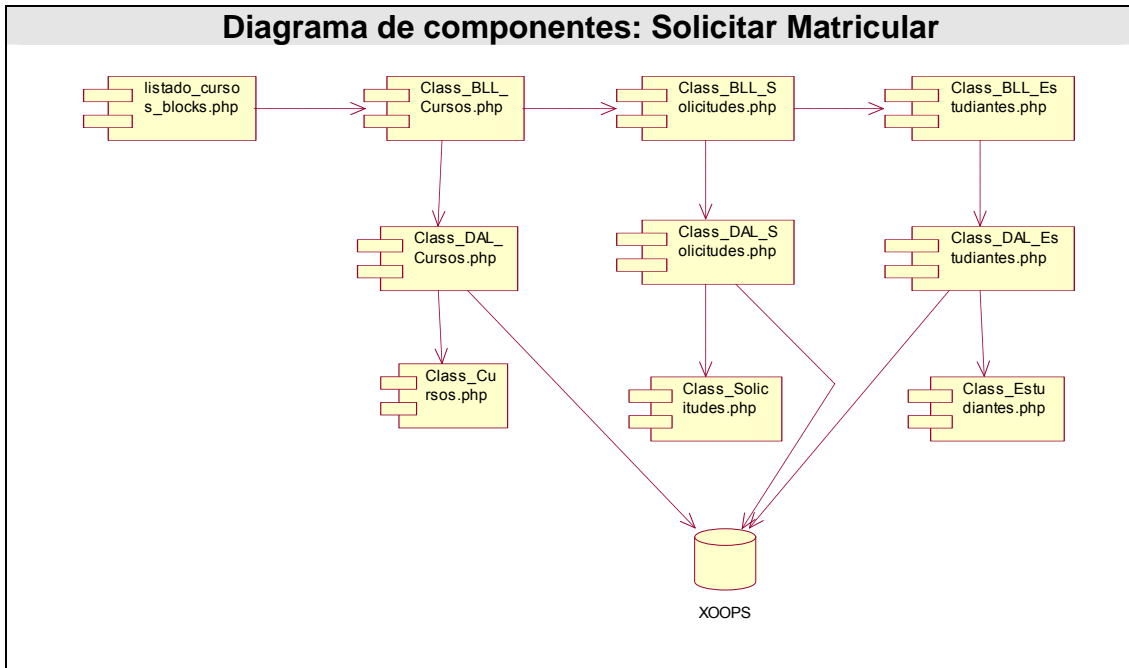
consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue. [25]

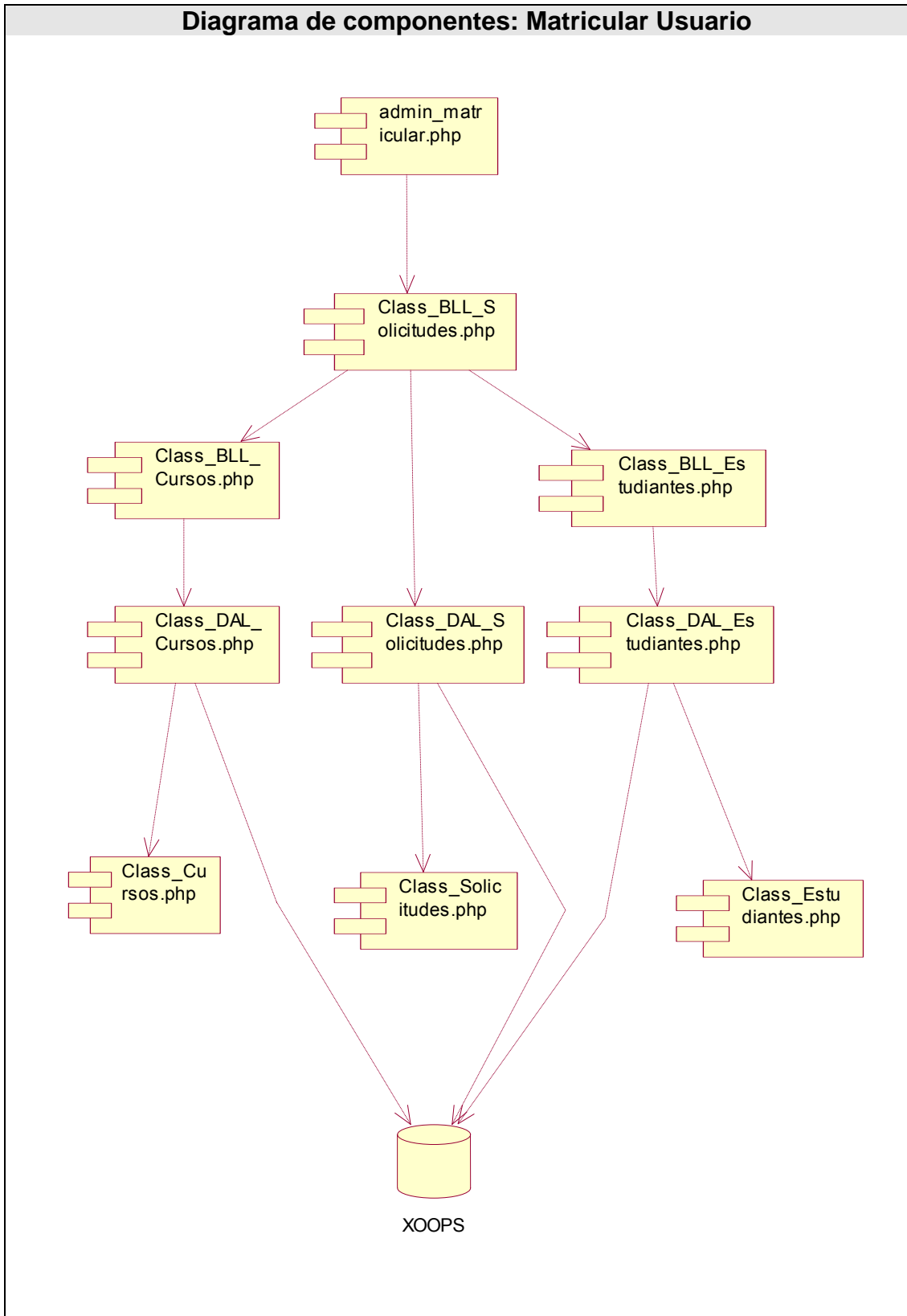
Decir que los paquetes que están de color verde pertenecen a Xoops y que dentro de Modules se encuentran más módulos pero solo nos centraremos en el de eLearning.

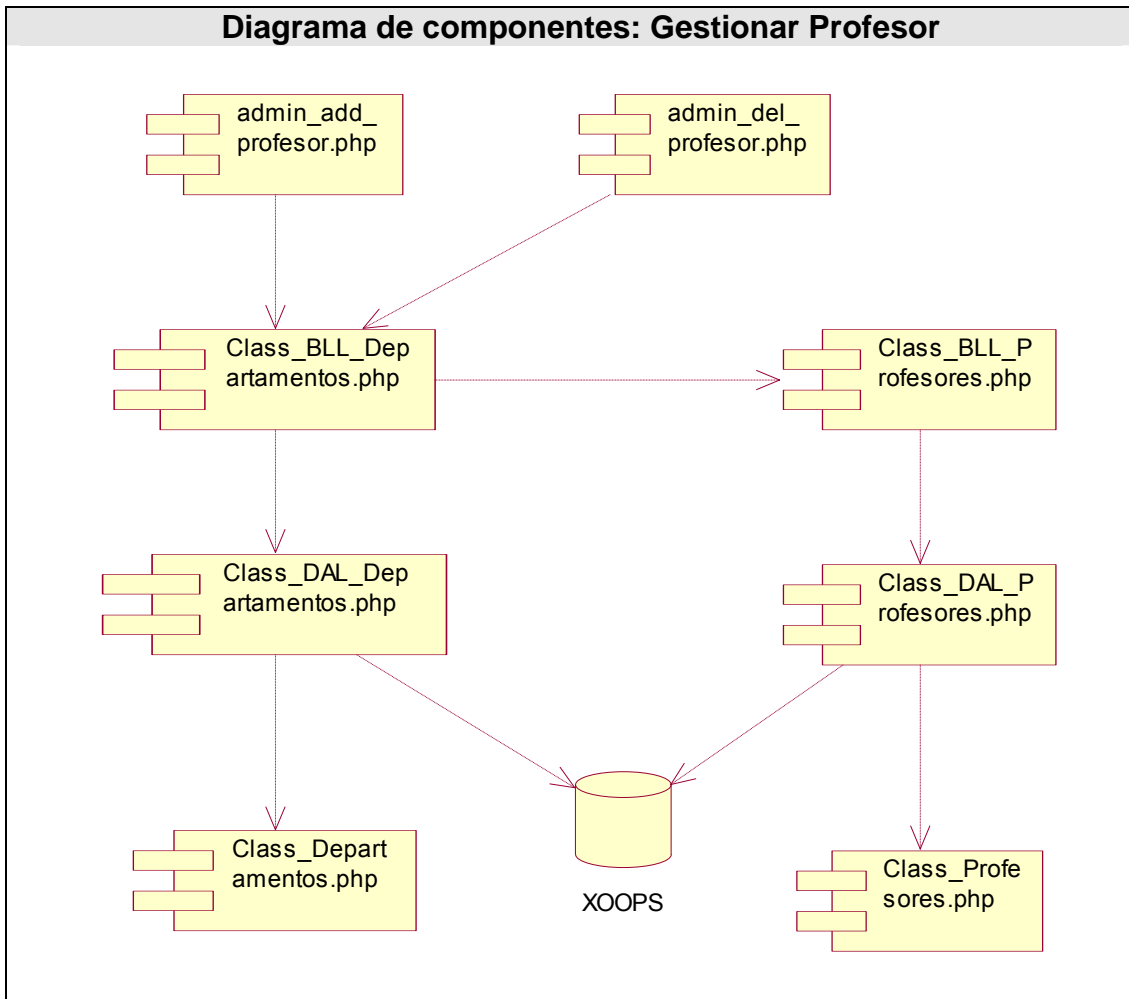


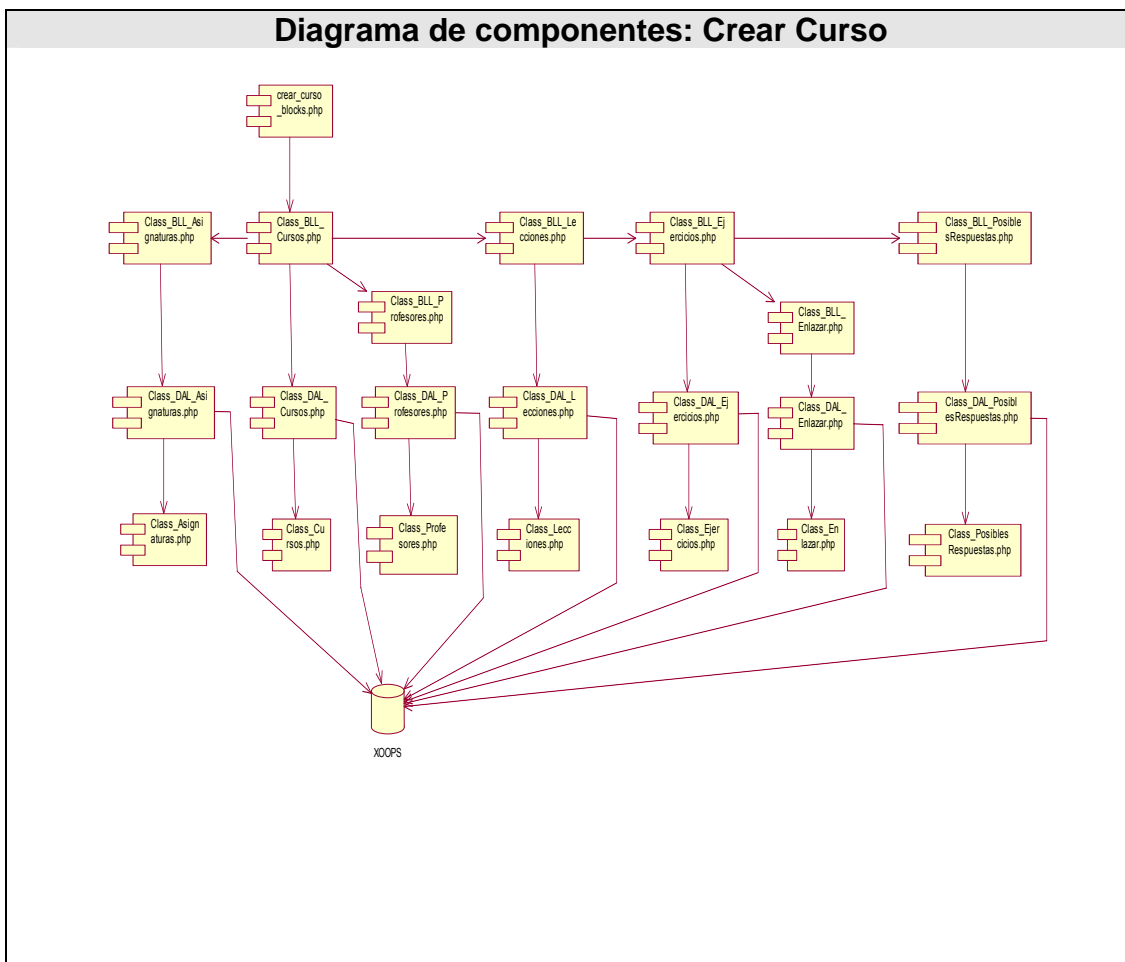
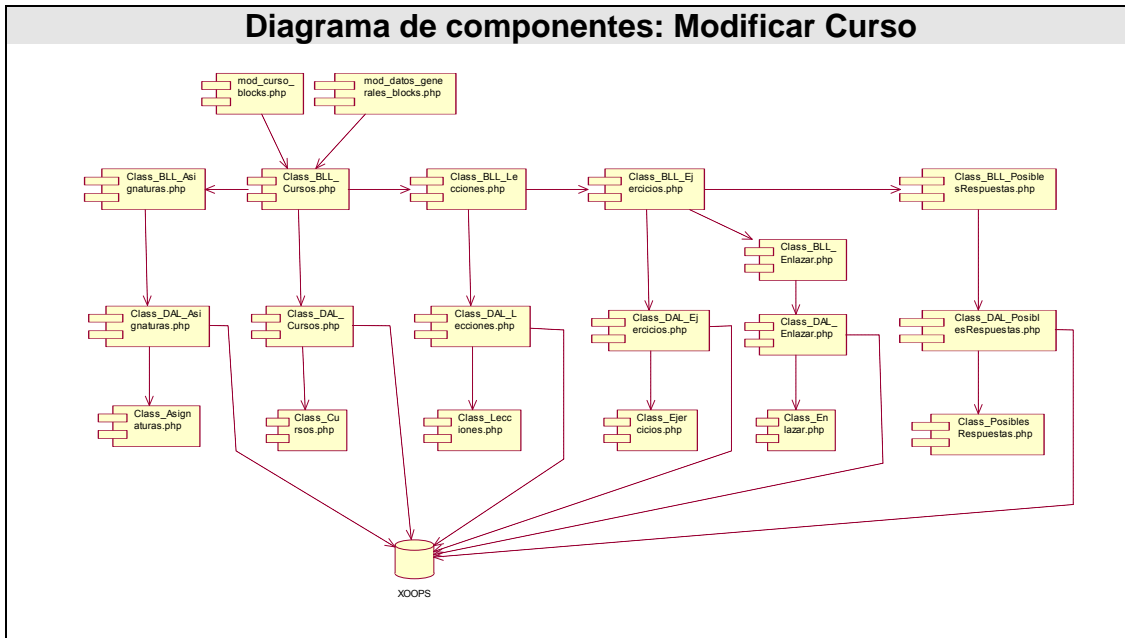












Componente	Propósito	Contenido	Interfaces
INDEX.php	Fichero de inicio.	-	-
Kernel	APIs del CMS.	-	-
Cache	Contiene los bloques y los ficheros temporales de los módulos.	-	-
Class	Aquí están todas las clases del CMS Xoops.	-	-
Language	Para cambiar el idioma del CMS Xoops.	-	-
Include	Aquí están todos los ficheros útiles que necesita el Xoops y que pueden ser útiles para los módulos.	-	-
Modules	Aquí dentro están todos los módulos que puede usar del CMS Xoops.	-	-
Xoops:db	Base de Datos de Xoops.	-	-
eLearning	Este es el módulo que desarrollamos.	-	-
index.php	Índice del módulo de administración.	-	-
menú.php	Menú para el módulo de administración.	-	-
admin_add_profesor_form.php	Genera el código HTML necesario para agregar un profesor.	-	-

admin_del_profesor_form.php	Genera el código HTML necesario para eliminar un profesor.	-	-
admin_matricular_form.php	Genera el código HTML necesario para matricular un usuario.	-	-
admin_add_profesor.php	Página de administración para agregar profesores.	Contiene los métodos para que el administrador del sistema pueda agregar un profesor.	Listar_Profesores() Adicionar_Profesor()
admin_del_profesor.php	Página de administración para eliminar un profesor.	Contiene los métodos para que el administrador del sistema pueda eliminar un profesor.	Listar_Profesores() Eliminar_Profesor()
admin_matricular.php	Página de administración para aceptar las matrículas de los usuarios.	Contiene los métodos para que el administrador del sistema acepte una matrícula.	Listar_Matriculas() Activar_Matriculas() Denegar_Matriculas()
listado_cursos_blocks.php	Página para listar los cursos.	Contiene los métodos para listar los cursos.	Listar_Cursos()
crear_cursos_blocks.php	Página para crear los cursos.	Contiene los métodos para crear un curso.	Guardar_Curso()
cursar_leccion_bloks.php	Página pasar un curso.	Contiene los métodos para pasar un curso	Cargar_Leccion()
estado_block.php	Para que el estudiante pueda ver sus "estadísticas" (La lección por la que va, Notas alcanzadas, etc.)	Contiene los métodos para mostrar las "estadísticas" del estudiante.	Cargar_Estados()
mod_curso_blocks.php	Página para modificar un curso.	Contiene los métodos necesarios para modificar un curso.	Modificar_Curso()
mod_datos_gene	Página para	Contiene los métodos	Modificar_Datos_Gene

enerales_bloques.php	modificar los datos generares de un curso.	necesarios para modificarlos.	rales()
listado_lecciones_blocks.php	Mostrar todas las lecciones del curso actual.	Contiene los métodos para mostrar las lecciones de un curso actual.	Listar_Lecciones()
Class_BLL_Asignaturas.php	Capa de negocio.	Class_BLL_Asignaturas	Insertar() Eliminar() Actualizar() Listado_Cursos()
Class_BLL_Cursos.php	Capa de negocio.	Class_BLL_Cursos	Insertar() Eliminar() Actualizar() Mostrar_Informacion_Generar_Curso() Listado_Lecciones() Listado_Estudiantes() Listado_Solicitudes_Por_Aprobar()
Class_BLL_Departamentos.php	Capa de negocio.	Class_BLL_Departamentos	Insertar() Eliminar() Actualizar() Listado_Profesores() Listado_Disciplinas() Adicionar_Disciplina() Eliminar_Disciplina()
Class_BLL_Disciplinas.php	Capa de negocio.	Class_BLL_Disciplinas	Insertar() Eliminar() Actualizar() Listado_Asignaturas() Departamento_AI_Que_Pertenece() Adicionar_Asignaturas()
Class_BLL_Ejercicios.php	Capa de negocio.	Class_BLL_Ejercicios	Insertar_Ejercicio() Eliminar_Ejercicio () Actualizar_Ejercicio () Insertar_Evaluacion() Eliminar_Evaluacion () Actualizar_Evaluacion () Listar_Ejercicio_Por_Tipo() Calificar_Ejercicio()
Class_BLL_E	Capa de	Class_BLL_Enlazar	Insertar()

nlazar.php	negocio.			Eliminar() Actualizar() Listado_Ejercicios()
Class_BLL_Estudiantes.php	Capa negocio.	de	Class_BLL_Estudiantes	Insertar() Eliminar() Actualizar() Listados_Estudiantes() Información_Estudiente()
Class_BLL_Lecciones.php	Capa negocio.	de	Class_BLL_Lecciones	Insertar() Eliminar() Actualizar() Listado_Lecciones() Listado_Ejercicios() Mostrar_Prueba() Esta_Pasada() Calificar_Prueba()
Class_BLL_PosiblesRespuestas.php	Capa negocio.	de	Class_BLL_PosiblesRespuestas	Insertar() Eliminar() Actualizar() Listado_Ejercicios()
Class_BLL_Profesores.php	Capa negocio.	de	Class_BLL_Profesores	Insertar() Eliminar() Actualizar() Listado_Profesores() Información_Profesor()
Class_BLL_Solicitudes.php	Capa negocio.	de	Class_BLL_Solicitudes	Insertar() Eliminar() Actualizar() Mostar_Solicitudes_Por_Aprobar() Aprobar_Solicitudes()
Class_DAL_Asignaturas.php	Capa acceso datos.	de a	Class_DAL_Asignaturas	Lo Mismo que Class_BLL_Asignaturas
Class_DAL_Cursos.php	Capa acceso datos.	de a	Class_DAL_Cursos	Lo Mismo que Class_BLL_Cursos
Class_DAL_Departamentos.php	Capa acceso datos.	de a	Class_DAL_Departamentos	Lo Mismo que Class_BLL_Departamentos
Class_DAL_Disciplinas.php	Capa acceso datos.	de a	Class_DAL_Disciplinas	Lo Mismo que Class_BLL_Disciplinas
Class_DAL_Ejercicios.php	Capa acceso datos.	de a	Class_DAL_Ejercicios	Lo Mismo que Class_BLL_Ejercicios

Class_DAL_Enlazar.php	Capa de acceso a datos.	de a	Class_DAL_Enlazar	Lo mismo que Class.BLL_Enlazar
Class_DAL_Estudiantes.php	Capa de acceso a datos.	de a	Class_DAL_Estudiantes	Lo mismo que Class.BLL_Estudiantes
Class_DAL_Lecciones.php	Capa de acceso a datos.	de a	Class_DAL_Lecciones	Lo mismo que Class.BLL_Lecciones
Class_DAL_PosiblesRespuestas.php	Capa de acceso a datos.	de a	Class_DAL_PosiblesRespuestas	Lo mismo que Class.BLL_PosiblesRespuestas
Class_DAL_Profesores.php	Capa de acceso a datos.	de a	Class_DAL_Profesores	Lo mismo que Class.BLL_Profesores
Class_DAL_Solicitudes.php	Capa de acceso a datos.	de a	Class_DAL_Solicitudes	Lo mismo que Class.BLL_Solicitudes
Class_Asignaturas.php	Clase persistente.		Class_Asignaturas	-
Class_Cursos.php	Clase persistente.		Class_Cursos	-
Class_Departamentos.php	Clase persistente.		Class_Departamentos	-
Class_Disciplinas.php	Clase persistente.		Class_Disciplinas	-
Class_Ejercicios.php	Clase persistente.		Class_Ejercicios	-
Class_Enlazar.php	Clase persistente.		Class_Enlazar	-
Class_Estudiantes.php	Clase persistente.		Class_Estudiantes	-
Class_Lecciones.php	Clase persistente.		Class_Lecciones	-
Class_PosiblesRespuestas.php	Clase persistente.		Class_PosiblesRespuestas	-
Class_Profesores.php	Clase persistente.		Class_Profesores	-
Class_Solicitudes.php	Lógica de negocio.	de	Class_Solicitudes	-

4.7. Conclusiones.

En el presente capítulo se ha descrito los principios del diseño para el desarrollo necesario de llevar a cabo el siguiente proceso de implementación del sistema. Donde se realizaron diferentes diagramas como: el diagrama de clase persistente que te permite realizar el diagrama de entidad-relación, los distintos diagramas de clase Web que te brinda una buena comprensión del modelo de negocio del sistema, también el diagrama de componente y el diagrama de despliegue.

Capítulo V.

Estudio de factibilidad.

5.1. Introducción.

En el presente capítulo se abordará el tema del estudio y factibilidad de sistema donde se brinda una descripción de la planificación del proyecto así como los costos asociados al mismo y los beneficios tangibles e intangibles que reportaría su elaboración. Finalmente, utilizando la Metodología COCOMO II, se realiza un análisis entre los costos y beneficios que aporta el sistema para llegar a la conclusión si es o no tangible el desarrollo del proyecto.

Se han introducido en la Ingeniería del Software una serie de técnicas, utilizadas dentro de las tareas de planificación, que ayudan a planificar y controlar el esfuerzo y el tiempo necesario de desarrollo. Para realizar estas estimaciones utilizamos el modelo de COCOMO (CONstructive COSt MOdel). COCOMO es una jerarquía de modelos de estimación de costos software que incluye submodelos básico, intermedio y detallado.

5.2. Planificación.

Para realizar el cálculo de los costos de desarrollo del sistema se deben obtener primero las instrucciones fuentes. Analizándose para esto las cantidades de entradas, salidas, peticiones, archivos lógicos que tiene el sistema.

Nombre de la entrada externa	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Crear Curso	6	46	Complejo
Crear Ejercicio	3	16	Complejo
Crear Lección	4	22	Complejo
Crear Evaluación	3	16	Complejo
Crear Profesor Virtual	2	5	Simple

Eliminar Profesor Virtual	1	7	Simple
Matricular Usuario	1	8	Simple
Modificar Lección	2	22	Complejo
Modificar Curso	6	33	Complejo
Modificar Ejercicio	3	10	Medio
Modificar Evaluación	3	10	Medio

Tabla 1 Entradas Externas

Nombre de la salida externa	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Listado de Departamentos	1	1	Simple

Tabla 2 Salidas Externas

Nombre de la petición	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Listado de Disciplinas	1	1	Simple
Listado de Asignaturas	1	1	Simple
Listado de Cursos	2	2	Simple
Listado de Lecciones	2	2	Simple
Mostrar la Lección	4	22	Complejo
Listado de Ejercicios	1	2	Simple
Mostrar un Ejercicio	3	7	Medio
Mostrar una Evaluación	3	7	Medio
Mostrar Estado	3	4	Simple

Tabla 3. Peticiones.

Nombre del fichero interno	Cantidad de records	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
xoops_asignaturas	1	3	Simple
xoops_curso_estudiante	1	4	Simple
xoops_cursos	1	6	Simple
xoops_departamentos	1	2	Simple
xoops_disciplinas	1	3	Simple
xoops_ejercicios	1	5	Simple
xoops_enlazar	1	6	Simple
xoops_estudiantes	1	8	Simple
xoops_lecciones	1	6	Simple
xoops_posibles_respuesta	1	5	Simple
xoops_profesores	1	7	Simple
xoops_solicitudes	1	7	Simple

Tabla 4. Ficheros Internos.

Elementos	Simples		Medios		Complejos		Subtotal de puntos de función
	No	X Peso	No	X Peso	No	X Peso	
Ficheros lógicos internos	11	7	0	10	0	15	77
Entradas externas	3	3	2	4	6	6	53
Salidas externas	1	4	0	5	0	7	4
Peticiones	6	3	2	4	1	6	32
Total							166

Tabla 5. Puntos de Función desajustados

5.3. Costos.

Características	Valor
Puntos de función desajustados	166
Lenguaje	PHP (85 %) JavaScript (3 %) HTML (12 %)
Instrucciones fuentes por puntos de función	(21)

	(56)
	(43)
Instrucciones fuentes por lenguaje (miles de instrucciones)	(2,9631)
	(0,2789)
	(0,8566)
Instrucciones fuentes (miles de instrucciones)	4,0986

Tabla 6. Cantidad de Instrucciones Fuentes.

Cálculo de:	Valor	Justificación
Esfuerzo	7,0811	SF_i PREC = 6,2 FLEX = 5,07 RESL = 5,65 TEAM = 0 PMAT = 3,12 $\sum SF_i = 20,04$ E = 1,1104 $\Pi EM_i \approx 0,5029$
Tiempo de desarrollo	13,91 meses	$PM \approx 7,0811$ $F \approx 0,6808$
Cantidad de personas	2	Se cuenta con 2 personas para la realización del sistema.
Costo	3186,50	$C = CHM * PM$ $CHM = 2 * 225 = 450$ $PM \approx 7,0811$
Salario medio	225	
RCPX	0,83	RELY = NOMINAL DOCU = NOMINAL CPLX = NOMINAL DATA = PEQUEÑO
RUSE	1,00	Existe concordancia entre la documentación y las necesidades del ciclo de vida.
PDIF	1,00	TIME = 50% STOR = 50% PVOL = 50%.

PREX	1,00	APEX = NOMINAL PLEX NOMINAL LTEX = NOMINAL
FCIL	0,73	TOOL = MUJY ALTO SITE = NOMINAL.
SCED	1,00	SCED = 130 %
PERS	0,83	ACAP = ALTO PCAP = ALTO PCON = ALTO

Tabla 7. Cálculos Finales de La Estimación de Costos y Esfuerzos.

5.4. Análisis de costos y beneficios.

El desarrollo de un proyecto informático tiene asociado un costo y realizarlo está independencia de los beneficios que el pudiera brindar. Los beneficios pueden ser de carácter social y económico. Ambos son de suma importancia.

La herramienta que se plantea tiene como objetivo fundamental a la educación y formación de estudiantes, por tanto su mayor beneficio es de carácter social. Esta aplicación que se quiere llevar acabo es de suma importancia ya que es unas de las herramientas que esta tomando un formidable auge en el desarrollo de la informática, en nuestro país se cuenta con muy pocas herramienta propia de esta índole.

El software reporta los siguientes beneficios para la formación:

- Permite al estudiante su formación educacional.
- Permite al estudiante cursar sus estudios de forma no presencial.
- Permite tanto al estudiante como al profesor ahorro de tiempo porque ninguno de los dos tiene que estar en un aula presencialmente para impartir los cursos y recibir los curso, así que pueden dedicar este tiempo a otras cosas (ejemplo: los estudiantes de la UCI que están inmerso en proyecto y no puede estar presencial mentes en los cursos por su dedicación a la producción de software).

5.5. Conclusiones.

En este capítulo se realizó el estudio de factibilidad correspondiente al sistema a desarrollar a través del modelo COCOMO. Es uno de los modelos más documentados en la actualidad y es muy fácil de utilizar. Aquí se hizo mención de algunos de los beneficios con el desarrollo de este módulo. Se realizó además un análisis entre el costo y los beneficios que reporta la aplicación concluyendo que es factible la aplicación.

Conclusiones generales.

El ambiente eLearning requiere de la toma de consideraciones de una serie de aspecto que permita que dichos espacios tenga lugar de lo que se espera de ellos: ¡Un aprendizaje eficiente! Para esto, son importantes no solo las herramientas que se usan, si no en la forma de cómo se usan las mismas.

Hoy en día las plataformas de formación no solo se están abriendo camino en las grandes empresas si no también en las diferentes instituciones educacionales donde se están adaptando a sus características educacionales pedagógicas. En nuestro país y en esta institución universitaria su objetivo fundamental es hacías las necesidades educativa y a las característica pedagógica seguida por la institución, que es con un solo fin la formación de estudiantes sin que se encuentre presencialmente en el aula.

Con este trabajo que permite agregar un módulo de eLearning dentro del portal de software libre del la UCI se alcanza el objetivo trazado por el desarrollo de este sistema, que me permita editar curso dentro del portal, en un área donde la formación del profesional constituye, hoy por hoy, uno de sus principales retos.

Recomendaciones.

Los objetivos trazados desde un inicio de este trabajo han sido logrados, pero en el transcurso de desarrollo de este sistema queda claro que la propuesta de este sistema es la primera fase de un proyecto que tiene la característica de ser más ambicioso en su evolución y desarrollo. Por tanto hacemos las siguientes recomendaciones:

- Ampliar más el sistema, no solo en un módulo de eLearning, sino a una plataforma de educación a distancia.
- Ampliar mas la base de datos para dar soporte a todos los módulos que integran una plataforma de educación a distancia.
- Hacer uso de los estándares internacionales.
- Hacer uso de una metodología pedagógica estándar para la realización de las plataformas de educación a distancias.
- Independizar el sistema, que no este incluido como un modulo dentro de un CMS sino que sea un LMS.
- Que el sistema tenga dentro todas las herramientas disponibles y necesarias que necesitan los usuarios para realizar las diferentes tareas dentro del sistema (Chat, foro de discusión, correo electrónico, etc.)

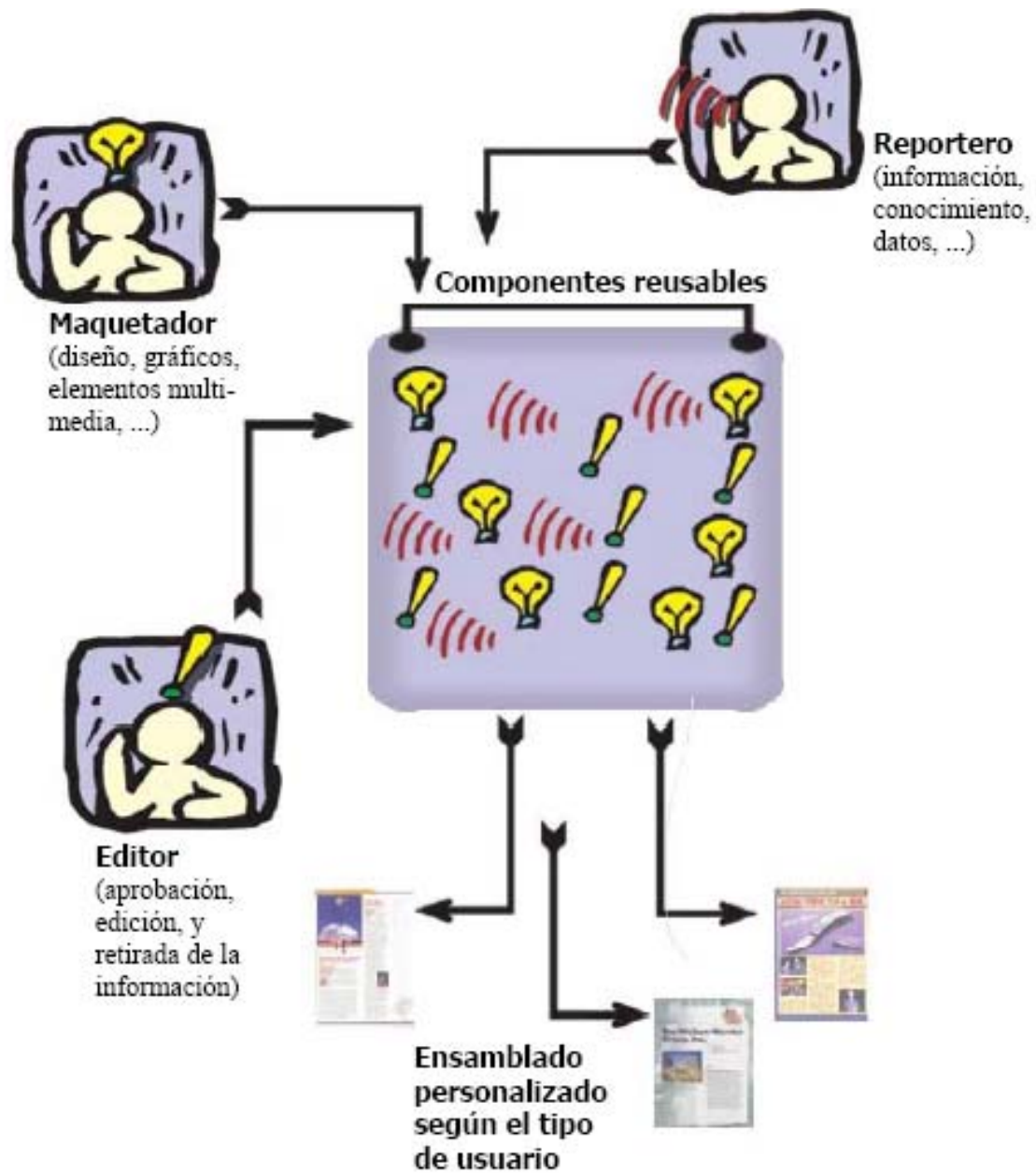
Bibliografías.

- [1]. Sistemas de Gestión de Contenidos: Artículo: "What is a content management system?". (06/2003). http://www.steptwo.com.au/papers/kmc_what/.
- [2]. Sistemas de Gestión de Contenidos: Artículo: "Wikipedia en español" <http://es.wikipedia.org/wiki/CMS>.
- [3]. Educación virtual: "Experto profesional. url" <http://www.uned.es/elearning/experto2005/experto/elearn/lms.html>.
- [4]. Sistemas de Gestión de Aprendizaje: "Educate Global" http://www.educate-global.com/mentorlms_quees.htm
- [5]. Sistemas de Gestión de Aprendizaje: "Educaterra" http://www.gt.educaterra.com/mgp/hojas/articulos/detallearticulo.jsp?articulo=4640&repositorio=148&pagina=1&idapr=21_789_esp_10.
- [6]. Sistemas de Gestión de Aprendizaje: "Portal Formativo" http://www.portalformativo.com/e-learning--ventaja-competitiva-de-las-instituciones-a_2_79.html.
- [7]. Sæther Bakken, Stig. Schmid, Egon. Manual de Php. PHP Documentation Group. 2003. <http://www.php.net/docs.php>.
- [8]. [Díaz, 2004] Díaz Antón Maria Gabriela, Angélica Pérez Maria "Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática" 2004. URL: <http://www.academia-interactiva.com/ise.pdf>, Mayo 2004.
- [9]. Booch, G., Rumbaugh, J., Jacobson, I. "El Lenguaje Unificado de Modelado". Addison-Wesley. 1999.
- [10]. [Ferrá, 2004] Ferrá Grau Xavier "Desarrollo orientado a objetos con UML Universidad Politécnica de Madrid, 2004 URL: <http://www.clikear.com/manuales/uml/introduccion.asp>, Mayo2004.
- [11]. [Letelier, 2004] Letelier Torres Patricio "Desarrollo de Software Orientado a Objeto usando UML, Departamento Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España, Mayo, 2004. URL: [URL:http://www.creangel.com/uml/intro.php](http://www.creangel.com/uml/intro.php).
- [12]. Larman, C. "Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design". Prentice-Hall, Inc. 1998.
- [13]. Joseph, Schmuller. "Aprendiendo UML en 24 horas", Prentice-Hall, Inc. 2001

- [14]. Sæther Bakken, Stig. Schmid, Egon. Manual de Php. PHP Documentation Group. 2003. <http://www.php.net/docs.php>.
- [15]. Sistemas de Gestión de Base de datos: “Base de datos en castellano”
<http://www.programacion.net/bbdd/tutoriales/idMySQL>.
- [16]. Sistemas de Gestión de Base de datos: “Arsys”
<http://www.arsys.es/soporte/productos/alojamiento/faqbasesdatos.htm#23>.
- [17]. Sistemas de Gestión de Contenidos: “Centro de Información, Investigación y Desarrollo de Soluciones de e-Learning ” <http://www.sceu.frba.utn.edu.ar/e-learning/introe-Ingtech.htm>
- [18]. [Jacobson, 2000] Jacobson, I.; Booch, G. y Rumbaugh, J.; “El Proceso Unificado de Desarrollo de software”. 2000. Addison-Wesley.
- [19]. Sistema de herramientas: “Cursos de informática gratuitos”
http://www.aulaclie.es/dreamweaverMX/f_dreamweavermx.htm.
- [20]. Procedimientos básicos para el desarrollar un proyecto en UML
<http://www.vico.org/TallerRationalRose.pdf>.
- [21]. Sistema de Gestión de Contenido: “XOOPS”
<http://www.esxoops.com/docu/newbies/index.html>.
- [22]. Tecnología de servidor PHP:” Maestros de Web”
<http://www.maestrosdelweb.com/editorial/zendstudio/>.
- [23]. Grupo:” UML” <http://grupo-uml.frcu.utn.edu.ar/apuntes/apunte2/ar01s02.html>.
- [24]. Modelo de Datos <http://www.ing.ula.ve/~ibc/modDatDepVjctfi.pdf>.
- [25]. Estudio sobre “UML” <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>

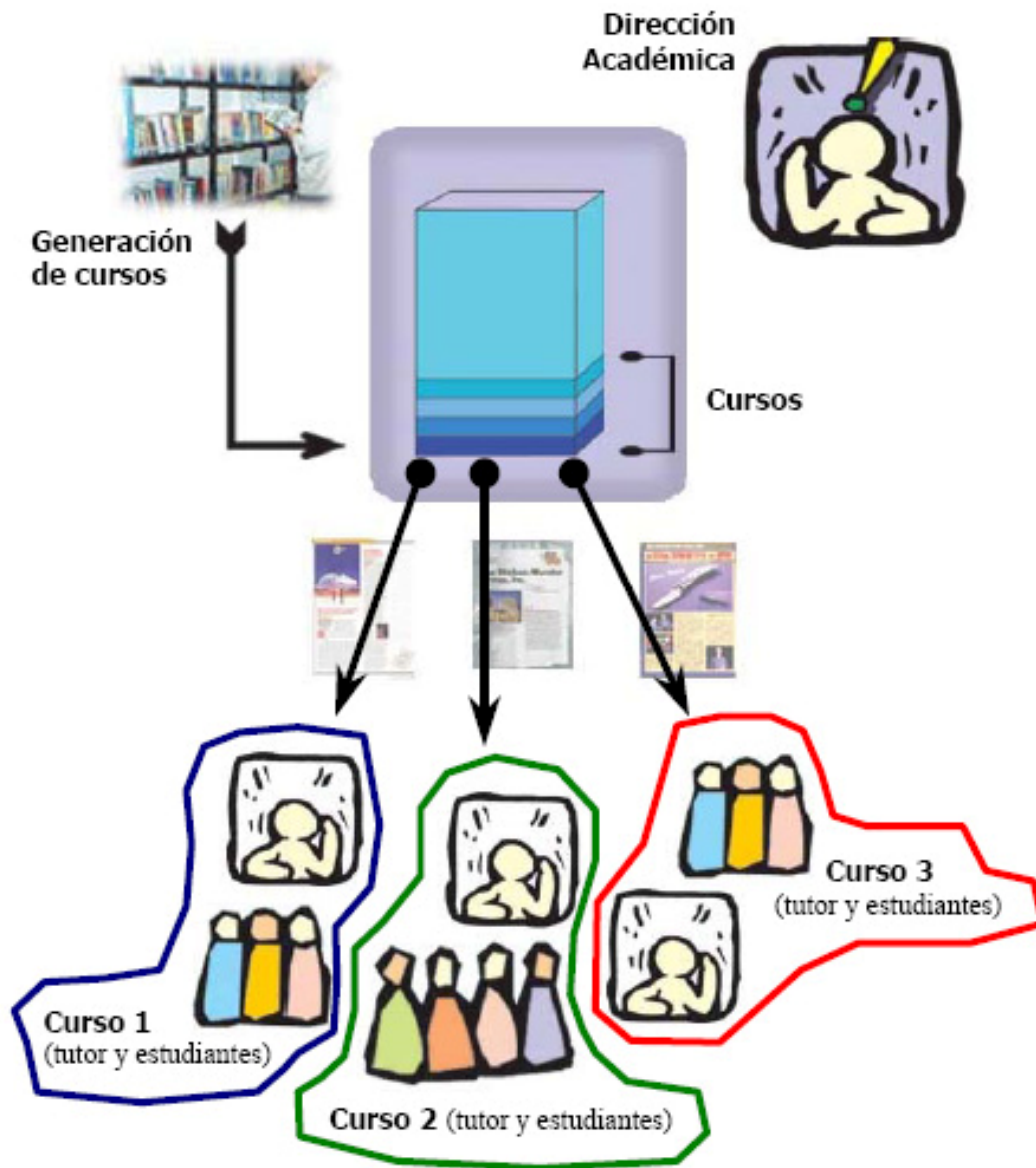
Anexos

Anexo 1



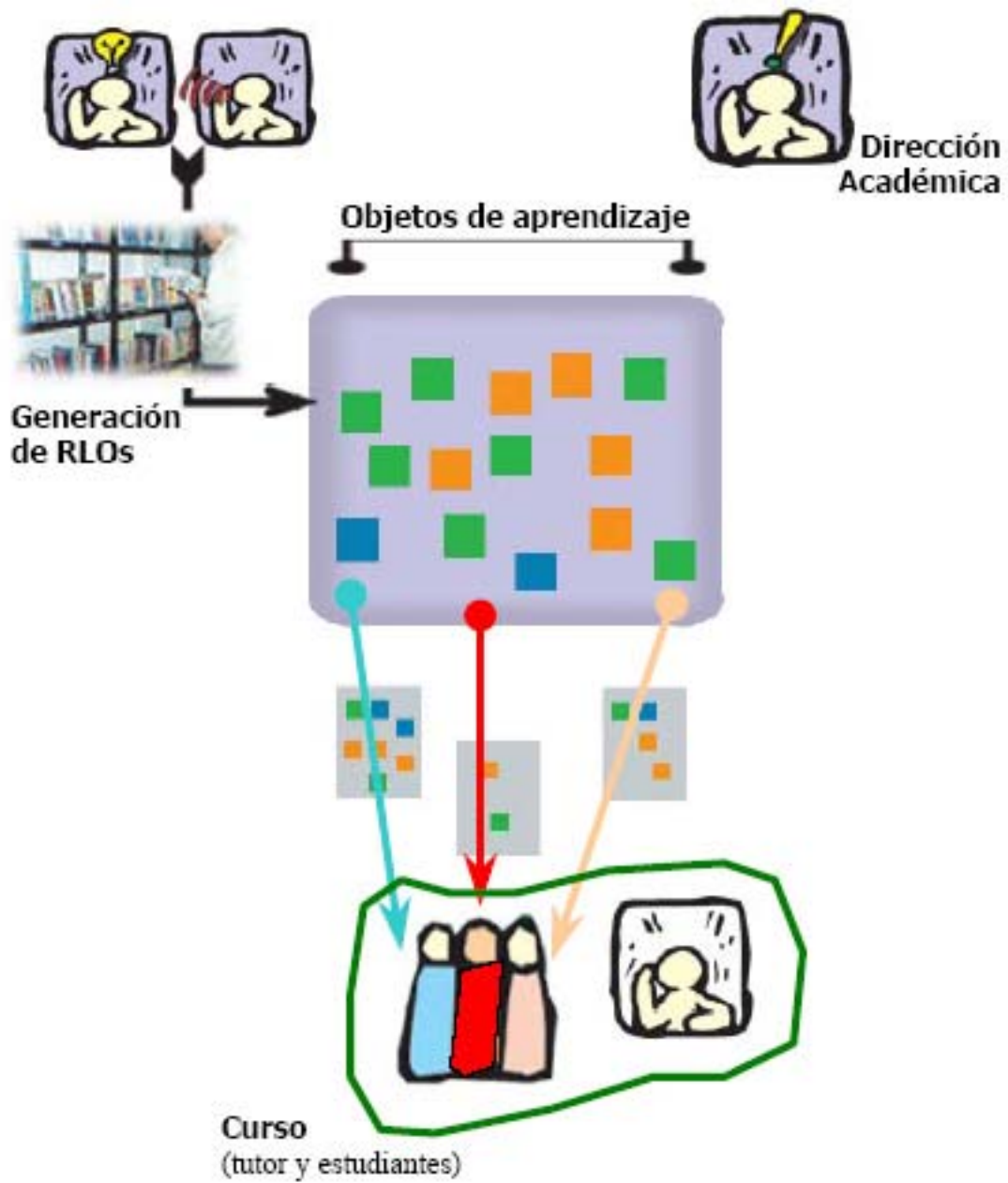
Esquema CMS (Content Management Systems)

Anexo 2



Esquema LMS (Learning Management Systems)

Anexo 3



Esquema LCMS (Learning Content Management Systems)

Glosario de términos.

Aprendizaje a distancia: El fin deseado de la educación a distancia.

Accesibilidad: Característica del diseño de una página Web. Los sitios accesibles pueden ser entendidos y navegados por las personas con alguna discapacidad.

Aplicación: Es el programa que el usuario activa para trabajar en el ordenador. Existen muchos programas de ordenador que pueden clasificarse como aplicación. Generalmente se les conoce como Software.

Aprendizaje asíncrono: Aprendizaje en el que la interacción alumno-profesor ocurre en forma intermitente y con retraso en el tiempo. Ejemplos de esto son los cursos de formación a través de Internet o CD ROM, tutorías pregunta respuesta, grupos de discusión en línea y correo electrónico.

Aprendizaje síncrono: Proceso de aprendizaje en línea, llevado a cabo en tiempo real, donde la interacción alumno - tutor coincide en el tiempo y en el espacio virtual.

Aprendizaje en línea: Aprendizaje provisto por tecnologías basadas en Web o basadas en Internet. Véase como Formación basada en Web y Formación basada en Internet.

Asesor: Profesional académico que, utilizando los elementos didáctico-pedagógicos en la conducción del aprendizaje, identifica las necesidades educativas del alumno y para satisfacerlas, lo apoya fomentando el autodidactismo y estimula la actitud analítica, crítica y constructiva.

Aula virtual: Espacio de aprendizaje en línea donde aprendices y tutores interactúan.

Código Abierto: Es una tendencia internacional del desarrollo de software que profesa la distribución del código junto a las aplicaciones, se rigen por licencias tales como GNU/GPL.

Educación a distancia: Situación educativa en la que el instructor y los alumnos están separados en el tiempo, el espacio o ambos. Los cursos de educación o formación a distancia son llevados a lugares remotos de forma síncrona o asíncrona, incluyendo correspondencia escrita, texto, gráficos, audio, cinta de video, CD Rom, formación en línea audio y video-conferencia, televisión interactiva y fax. La educación a distancia no excluye el aula tradicional. La definición de Educación a Distancia es más amplia que la de e-Learning

E- Learning: Incluye una amplia gama de aplicaciones y procesos, tales como aprendizaje basado en la red, en el ordenador, aulas virtuales, cooperación digital. Incluye la entrega de contenidos vía Internet, extranet, intranet, (LAN/WAN), audio y vídeo, emisión satelital, televisión interactiva y CD-ROM.

HTTP: Es el conjunto de reglas para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

Teleformación: Es la formación a distancia utilizando las líneas telefónicas para la transmisión de voz, entre personas, o bien utilizando las líneas de datos para establecer la comunicación entre ordenadores.

Zend: Compañía líder de infraestructuras para Web, está reconocida internacionalmente como la autoridad actualmente en PHP. Sus fundadores son los diseñadores del PHP v.4 en adelante, actualmente es una compañía líder dentro de la comunidad Open Source.