

**Universidad de las Ciencias Informáticas
Facultad 1**



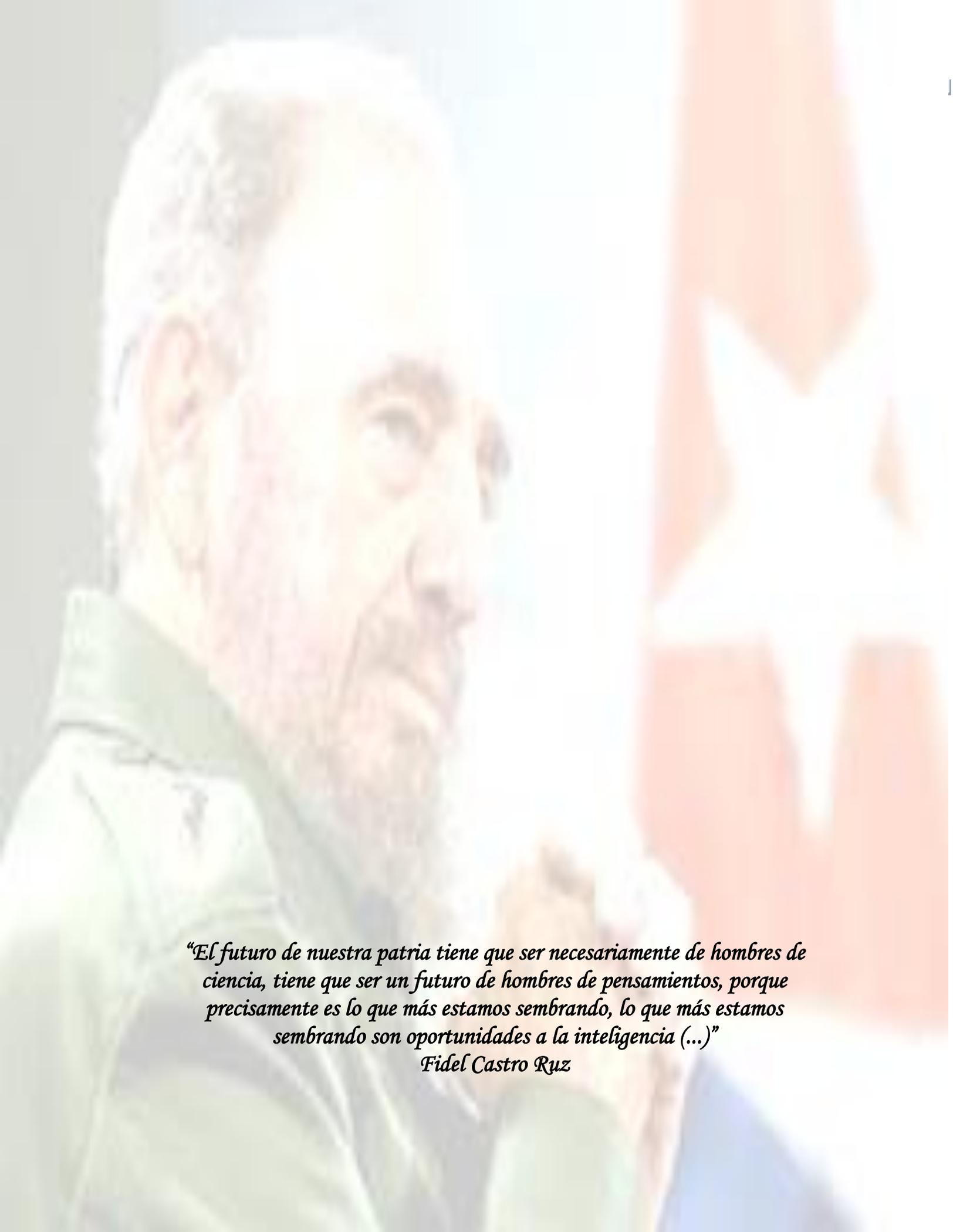
TÍTULO: Diseño e implementación de un *OpenCourseWare* sobre la arquitectura Drupal.

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

***AUTORES: ROXANA LARREA ESTRADA
ARIEL FAJARDO DELGADO***

***TUTORES: Ing. Maikel Manuel Fernández Fernández
Ing. Emilio Surí López***

LA HABANA, JUNIO DE 2014



“El futuro de nuestra patria tiene que ser necesariamente de hombres de ciencia, tiene que ser un futuro de hombres de pensamientos, porque precisamente es lo que más estamos sembrando, lo que más estamos sembrando son oportunidades a la inteligencia (...)”

Fidel Castro Ruz

Declaración de autoría

Declaramos ser autores de la presente tesis que tiene por título Diseño e implementación de un *OpenCourseWare* sobre la arquitectura Drupal y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Roxana Larrea Estrada

Firma del Autor

Ariel Fajardo Delgado

Firma del Tutor

Maikel Manuel Fernández Fernández

Firma del Autor

Emilio Surí López

Firma del Tutor

Agradecimientos

El largo trayecto recorrido en nuestra formación como estudiantes culmina con la obra cumbre, el trabajo de diploma. Pero para llegar a la cúspide son necesarios muchos sacrificios, no solo los propios, también ajenos. Todo el esfuerzo y las horas de empeño para lograr realizar este trabajo queremos dedicarlos a estas personas que han sido y son la razón de nuestro ser.

Roxana: *A mi mamá, que es lo más grande y hermoso que tengo en este mundo, por ser esa persona que nunca se dio por vencida conmigo a pesar de ser una niña un poco malcriada, por ser mi motor impulsor, por sacar fuerzas de donde no hay para seguir incitándome, por poner toda su vida y más en función de mi felicidad, por ser mi ejemplo de superación en la vida profesional y en esa larga fiesta de experiencias que es la vida, en fin por ser mi orgullo. Te agradezco tu dedicación, tus horas de desvelo, tu sabiduría y por qué no tus regañones, pues son los que me han llevado a ser la persona que soy hoy. **A mi papá,** por estar siempre pendiente de mí, por todo su cariño, porque sé que hoy se siente orgulloso de mí y de todo lo que he logrado. **A mis tutores Maikel y Emilio,** que a pesar de su apretada agenda de trabajo siempre estuvieron ahí disponibles para ayudarnos, sin ustedes no hubiera sido posible la realización de esta tesis, gracias por su apoyo. **A mi compañero de tesis,** por ayudarme a hacer realidad este sueño, por cada palabra de aliento, por siempre confiar en mí... el destino quería que celebráramos este día juntos, a pesar de los contratiempos logramos seguir como dúo de tesis. **A toda mi familia,** que a pesar de la distancia siempre se preocuparon por saber que todo me iba bien y que tuvieron plena confianza en que podría lograr este sueño. **A mi grupo 1503,** con el que tuve el placer de compartir innumerables momentos no pensé conocer jamás personas tan especiales como ustedes. **A amigas tan especiales** como Maide, Maylín, Ilo, Yanet y Mayrelis que supieron sobrellevar este carácter complicado que tengo y me dieron ánimos para continuar aún cuando las cosas se ponían difíciles, todas forman ya parte de mi vida, de cada una de ustedes me llevo lo mejor. **A Frank y Marlon,** dos de los amigos de los cuales recibí más apoyo en todos estos meses, por siempre creer en que podía alcanzar esta meta y por estar siempre ahí. **A todas esas amistades** con las cuales compartí durante estos 5 años, a los que ya no están aquí, a los que fui conociendo en cada curso, a los de otras facultades, a los de la facultad 1, a los del laboratorio de proyecto, a todas muchas gracias por brindarme el enorme placer de su amistad. **A los profes** que tuvieron que ver con mi formación como Ingeniera, en especial al profe Raydel Zumeta por acceder a atendernos y asesorarnos en la realización de este trabajo. A todas esas personas que en algún momento me preguntaron ¿Cómo va esa tesis?*

Ariel: *A mis padres y mi hermano por brindar todo su apoyo y dedicación a todas las decisiones que eh tomado durante toda mi vida. **A mi compañera de tesis** por tener paciencia y no rendirse jamás con este cascarrabias y sus caprichos. **A mis tutores** por su apoyo, dedicación y confianza en estos meses de trabajo. **A la Universidad y su colectivo de profesores** que me han formado durante estos cinco años. **A todas las amistades** que de alguna forma han contribuido a que llegara a la meta final. En fin, quiero agradecer **a todas aquellas personas** que de una forma u otra han contribuido al desarrollo de este trabajo de diploma y a mi formación tanto profesional como personal.*

Dedicatoria

A mi mamá por hacer de este también su sueño y poner todo su empeño en ayudarme a hacerlo realidad, porque nunca podré agradecerle lo suficiente todo lo que ha hecho por mí.

Roxana Larrea Estrada

A mis padres y familia por darme la educación que me ha permitido llegar hasta aquí, por formarme como un hombre independiente, por sus desvelos y por todos los sacrificios que ha tenido que hacer para que nunca me faltara nada y pudiera estudiar siempre sin ninguna preocupación. A mi hermanito para que siga mi ejemplo y se convierta en un gran profesional también.

Ariel Fajardo Delgado

Resumen

En la actualidad las acciones que se realizan en el proceso editorial se nutren de varias herramientas que mejoran la calidad y propician una mejor utilización del tiempo y los recursos. La modernización del sistema educativo está permitiendo acometer nuevas misiones a través de sistemas que contribuyen a la difusión y construcción pública del conocimiento, aportando un bien social. La presente investigación tiene como propósito describir y analizar la implementación de un sitio web *OpenCourseWare* basado en la arquitectura Drupal que permitió gestionar el flujo de trabajo editorial y publicar sus contenidos en formato digital. El documento integra un estudio de sistemas homólogos que proyecta importantes resultados sobre las características comunes en el área de investigación. Se realizó un estudio alrededor de las tendencias actuales en la creación de *OpenCourseWare* el cual arrojó un conjunto de características ineludibles en este tipo de sistemas y cómo aplicar Drupal para alcanzarlas. Para el desarrollo de la propuesta de solución se utilizó el Sistema de Gestión de Contenidos Drupal en su versión 7.24 y PostgreSQL versión 9.1 como Sistema Gestor de Bases de Datos. Además, se expone el proceso de desarrollo guiado por la metodología OpenUP a partir de la cual se generaron un conjunto de artefactos que resumen la información relacionada con la solución. Para verificar el cumplimiento de las funcionalidades propuestas se realizaron varias pruebas de *software* al sistema. Como resultado se obtuvo un producto funcional con una documentación que sirve de base para futuras investigaciones o modificaciones a la propuesta de solución.

Palabras claves: conocimiento, *OpenCourseWare*, proceso editorial, sistema educativo.

Índice

Introducción	1
Capítulo 1: Fundamentación teórica acerca del dominio conceptual y tecnológico.	6
1.1 Introducción	6
1.2 Definiciones importantes.	6
1.3 Estado actual y perspectivas del Movimiento de Acceso Abierto.	7
1.5 Iniciativas de trabajo colaborativo en el mundo.	9
1.6 OpenCourseWare en el mundo. Análisis de sistemas homólogos.	13
1.7 Descripción del entorno tecnológico para la creación de un OpenCourseWare.	17
Capítulo 2: Características de la propuesta de solución	30
2.1 Introducción	30
2.2 Propuesta del OpenCourseWare basado en la arquitectura Drupal.	30
2.3 Modelo de dominio.	30
2.4 Levantamiento de requisitos.	32
2.5 Modelo de Casos de Uso del sistema.	34
2.6 Modelo de diseño.	42
2.7 Diagramas de interacción.	46
2.8 Arquitectura del sistema.	47
2.9 Descripción de patrones utilizados.	48
2.10 Diseño de la base de datos.	50
2.11 Modelo de despliegue.	51
Capítulo 3: Implementación y prueba del sistema	53
3.1 Introducción	53
3.2 Implementación.	53
3.3 Estándares de codificación.	55
3.4 Pantallas de la aplicación.	56
3.5 Validación del sistema.	56
Conclusiones	64
Recomendaciones	65
Bibliografía consultada	69
Glosario de términos	72
Anexos	73

Índice de tablas

Tabla 1: Elementos Dublin Core.....	12
Tabla 2: Actores del sistema.....	35
Tabla 3: Especificación del caso de uso Gestionar Cursos.....	38
Tabla 4: Especificación del caso de uso Gestionar Categorías.....	41
Tabla 5: Descripción de la tabla public.Cursos.....	51
Tabla 6: Descripción de componentes.....	54
Tabla 7: Caso de prueba Gestionar curso.....	58
Tabla 8: Caso de prueba Gestionar categoría.....	59
Tabla 9: Especificación del caso de uso Gestionar materiales.....	76
Tabla 10: Especificación del caso de uso Gestionar boletín.....	77
Tabla 11: Especificación del caso de uso Gestionar Lista de cursos.....	79
Tabla 12: Especificación del caso de uso Gestionar artículos.....	81
Tabla 13: Caso de prueba Gestionar materiales.....	84
Tabla 14: Caso de prueba Autenticar usuario.....	85

Índice de figuras

Figura 1: <i>Revistas científicas en acceso abierto</i>	8
Figura 2: <i>Usuarios de OCW</i>	9
Figura 3: <i>Ciclo de vida de OpenUP (FRAMEWORK, 2013)</i>	22
Figura 4: Diagrama del modelo de dominio.....	31
Figura 5: Diagrama de casos de uso del sistema.....	36
Figura 6: Prototipo de interfaz de usuario del caso de uso Gestionar Cursos.....	39
Figura 7: Prototipo de interfaz de usuario del caso de uso Gestionar Categorías.....	42
Figura 8: Diagrama de clases del diseño web de Drupal.....	43
Figura 9: Diagrama de clases del diseño de Gestionar Curso.....	45
Figura 10: Diagrama de clases del diseño de Gestionar Categorías.....	45
Figura 11: Diagrama de secuencia RF Agregar curso.....	46
Figura 12: Diagrama de secuencia RF Editar curso.....	47

Figura 13: Arquitectura de Drupal (DRUPAL, 2011).....	47
Figura 14: Modelo de datos del sistema.	50
Figura 15: Diagrama de despliegue.....	51
Figura 16: Diagrama de componentes.....	54
Figura 17: Pantalla principal del sistema.	56
Figura 18: Vista de un curso creado.....	56
Figura 19: Cantidad de no conformidades por iteración.	59
Figura 20: Interfaz gráfica de la herramienta de seguridad Acunetix.....	60
Figura 21: Resultados de la primera iteración de pruebas de seguridad.....	60
Figura 22: Resultados de la segunda iteración de pruebas de seguridad.	61
Figura 23: Reporte resumen generado por la herramienta JMeter.	62
Figura 24: Prototipo de interfaz de usuario del caso de uso Gestionar Materiales.	74
Figura 25: Prototipo de interfaz de usuario del caso de uso Gestionar lista de cursos.	74
Figura 26: Diagrama de clases del diseño Gestionar artículos.....	82
Figura 27: Diagrama de clases del diseño Gestionar materiales.	82
Figura 28: Diagrama de secuencia RF Editar Categoría	83
Figura 29: Diagrama de secuencia RF Gestionar campos.	83

Introducción

En las sociedades contemporáneas la búsqueda constante de información marca nuevas pautas en las relaciones entre los individuos y aporta elementos sustanciales que generan nuevos conocimientos. Sin embargo, en sus inicios no se contaba con mecanismos para almacenar y brindar acceso público a gran escala al conocimiento. “No es hasta 1969, con la aparición de Internet”(MUSEUM, 2014), que se logra una infraestructura tecnológica que permite el flujo global de información borrando las barreras regionales existentes hasta la eclosión de la misma.

Lo que comúnmente hoy se denomina nuevas tecnologías vislumbran un fuerte arraigo en los hábitos y costumbres del hombre actual. En este sentido, las tecnologías que viabilizan y facilitan el flujo y transmisión de información han ganado un rol fundamental. El estudio de su esencia y el análisis de las consecuencias de su uso suscitan una disertación minuciosa por parte de diferentes disciplinas, las cuales arrojan varias situaciones, algunas en vías de solución.

Las tecnologías de la información y las comunicaciones (TIC o bien NTIC para nuevas tecnologías de la información y de la comunicación) agrupan los elementos y las técnicas usadas en el tratamiento y la transmisión de las informaciones, principalmente de informática, internet y telecomunicaciones (PÉREZ, 2011).

El uso de las TIC tiene una gran influencia en la vida de las sociedades contemporáneas, en las relaciones interpersonales y en la forma de difundir y generar conocimientos. Además, los continuos avances científicos y el escenario, matizado por las tendencias de la globalización económica y cultural, contribuyen a que los conocimientos sean perecederos y que se agilice la dinámica social, lo que entraña cambios en las estructuras económicas y culturales de la pluralidad contextual. No obstante, muchos sectores se nutren de los beneficios que tributan estas tecnologías. Las editoriales son de las favorecidas ya que estos adelantos contribuyen a la socialización de los resultados de su labor.

Los últimos diez años fueron testigos también de enormes transformaciones tecnológicas en la industria editorial. Dicha revolución ha alcanzado todos los aspectos de la actividad editorial, que ha pasado de ser una actividad exclusiva de ciertas instituciones con grandes recursos -a pesar del proceso de monopolización intenso que ocurre actualmente entre las grandes editoriales- a una actividad que la gran mayoría puede realizar con bastante independencia (TARRAGÓ, 2007).

En el sector editorial, el mayor impacto de estas tecnologías se asocia a la digitalización que permite la administración de contenidos digitales, la creación de flujos de trabajo digital distribuido y los procesos de producción controlados por medio de información digital.

Las TIC revolucionan todos los procesos editoriales previos a la impresión del original, así como los de impresión, distribución y comercialización de las publicaciones. Además, desde sus orígenes, aquellas

han posibilitado la creación y edición de nuevos productos como libros y revistas electrónicas, bases de datos a texto completo, multimedias, entre otros. Sin embargo, no solo propiciaron la aparición y desarrollo de las revistas electrónicas, sino también en otras plataformas de comunicación e intercambio, archivos electrónicos o repositorios institucionales que pretenden favorecer el acceso abierto a la literatura científica.

De igual forma, el resultado del trabajo intelectual de investigadores en todo el mundo era controlado por las casas editoras de revistas de divulgación científica debido a que son la herramienta para poder dar a conocer los resultados de su trabajo. Como consecuencia la difusión del conocimiento se vio limitada tanto por los altos costos que involucra la publicación tradicional, como por la monopolización antes mencionada.

De acuerdo a estas limitaciones eclosionan los movimientos de acceso abierto al conocimiento los cuales “pretenden eliminar barreras económicas y legales en la difusión de los resultados de la investigación. Su objetivo es maximizar el impacto de la investigación facilitando su acceso” (GUZMÁN, 2006).

Este movimiento se apoya en la definición **BBB**: Declaraciones de **B**udapest (BOAI, 2002), **B**erlín (2003) y **B**ethesda (2003) que rigen el funcionamiento del Modelo de acceso abierto. Estas “proponen la distribución de los contenidos con derecho a ser citados, reproducidos y compartidos de forma libre” y “el uso de herramientas de *software* libre para la gestión, preservación y distribución de los contenidos” siempre reconociendo el autor de un trabajo.

El modelo de acceso abierto se basa en dos estrategias fundamentales para garantizar el acceso sin restricciones económicas y legales de la información científico-técnica: las revistas de acceso abierto y los repositorios temáticos e institucionales. Sin embargo, en el caso de los repositorios, mayormente lo que registran son tesis y trabajos de diploma y la documentación que se genera al realizar estos trabajos no es almacenada en ninguna parte.

De igual forma aparecen un conjunto de iniciativas, propuestas y declaraciones sobre acceso abierto para el conocimiento que avalan la necesidad de la disponibilidad en Internet de la información científica. Por tanto, cada día se realizan más publicaciones de acceso abierto en revistas dirigidas a ello, y ante la disminución de problemas, son más los lectores que pueden acceder a ellas y mayor es el impacto de esas publicaciones; entre las que se encuentran los repositorios, las revistas científicas, las bibliotecas digitales, los sitios web *OpenCourseWare*(OCW) y los congresos.

Para la implementación y desarrollo de estas publicaciones se requiere de la interrelación de varias tecnologías que cubran distintos ámbitos de funcionamiento. De cualquier manera, las tecnologías específicas pueden ser varias e implementarse en distinta forma y para ello se necesita un marco tecnológico que resuelva y articule los requerimientos particulares de la institución que lo establece, así como las adaptaciones necesarias y específicas de cada comunidad académica. En este sentido, los

gestores de contenidos se presentan como una de las tecnologías empleadas para la organización, representación y distribución de la documentación destacándose Drupal como el gestor que más posibilidades ofrece para implementar este tipo de iniciativas de este modelo, gracias a su flexibilidad y modularidad. El principal inconveniente es la falta de una apuesta clara por éste, con un desarrollo específico para OCW pues se desarrollan muchos proyectos, pero se aboga por tener construidas estructuras que permitan de forma rápida desplegar un OCW.

Unido a esto, el proceso de recuperación de información se encuentra hoy plagado de inconsistencias que atentan contra su organización. Hoy existen plataformas virtualizadas para la educación a distancia que incluyen cursos con su estructura y forma de evaluación pero que presentan limitaciones en cuanto a la forma de su distribución. Estas plataformas, denominadas hoy sistemas de aprendizaje electrónico o sistemas *E-learning* se presentan como soluciones para optar por cursos como plataforma educativa pero sin un enfoque editorial.

En la Universidad de las Ciencias Informáticas se necesita fortalecer la educación que se brinda a la comunidad de estudiantes y profesores. Existen muchos materiales desplegados con este fin, pero muchas veces no se organizan en una estructura de cursos debido a que el acceso abierto dentro de la comunidad está pensado para recursos académicos y artículos científicos, no así para cursos u otros materiales didácticos lo que provoca inconvenientes en la recuperación de información y que se pierdan los materiales que se utilizan para construir conocimiento. Por consiguiente, se exige más tiempo de las personas, al estar dispersos y sin seguir un patrón específico. Por otra parte, en esta institución se emplean varias tecnologías por lo que los resultados deben responder a estándares de desarrollo de sistemas *OpenCourseWare* (OCW) que estén integrados bajo el protocolo OAI-PMH (de las siglas en inglés *Open Archives Initiative - Protocol Metadata Harvesting*) para evadir limitaciones en cuanto a su distribución.

La presente investigación surge por la necesidad de dar solución a la situación problemática anteriormente expuesta, a partir de la misma se plantea como **problema de la investigación**: ¿Cómo facilitar la gestión de materiales digitales dispersos, bajo un enfoque editorial y con fines formativos y didácticos?

EL **objeto de estudio** para dar solución a este problema, está enfocado hacia el modelo de acceso abierto a los contenidos académicos.

El **campo de acción** queda enmarcado en: los sistemas *OpenCourseWare*.

Dadas las condiciones mencionadas anteriormente se propone como **objetivo general de la investigación**: Desarrollar un *OpenCourseWare* sobre la arquitectura del Sistema de Gestión de Contenidos Drupal para facilitar la gestión de cursos en acceso abierto.

Para dar alcance al objetivo planteado se definieron como **objetivos específicos** los siguientes:

- Construir el marco teórico conceptual de la investigación que permita la comprensión del

escenario.

- Diseñar las entidades de gestión y sus relaciones.
- Implementar los procesos de negocio sobre la arquitectura del sistema de gestión de contenidos Drupal.
- Personalizar la presentación del sistema de gestión de contenidos Drupal.
- Validar los resultados del proyecto mediante pruebas de funcionalidad.

Se plantea como **idea a defender** que con el desarrollo de un *OpenCourseWare* se podrán organizar bajo un enfoque editorial y con fines formativos y didácticos, materiales digitales dispersos que permitan agilizar el proceso de recuperación de información.

Para lograr los objetivos propuestos se precisan las siguientes **tareas de la investigación**:

- Descripción de las características de un *OpenCourseWare*.
- Definición de la estructura de los cursos.
- Definición del perfil de usuario.
- Descripción del flujo editorial.
- Implementación de las estructuras de gestión.
- Implementación de los mecanismos de recuperación de información.
- Implementación de un proveedor de datos OAI-PMH.
- Realización de pruebas de funcionalidad.
- Valoración del impacto de la solución implementada.

Como **resultados esperados**:

- Un sitio web *OpenCourseWare* que cuente con estructuras que ofrezcan usabilidad, accesibilidad y funcionalidad.

Para asistir y dar alcance a las tareas propuestas se emplearon los siguientes **métodos científicos**:

- Analítico – sintético: método teórico que permitió analizar las características del movimiento de acceso abierto al conocimiento y las iniciativas principales de trabajo colaborativo que este abarca, así como las relaciones entre estas.
- Sistematización: Permitted desarrollar el objeto de la investigación mediante la integración de herramientas metodológicas que tributaron a la aplicación del modelo de acceso abierto a los contenidos académicos.
- Análisis histórico - lógico: este método tuvo como finalidad analizar la trayectoria completa del OCW, su condicionamiento a los distintos periodos evolutivos, las principales etapas por las que atravesó y las conexiones históricas fundamentales poniendo de manifiesto la lógica interna de su desarrollo.

- Observación: método empírico cuya función primaria es analizar la situación real de la presente investigación y percibir cómo se realiza la gestión de materiales digitales dispersos para el diagnóstico acerca de la realización del sistema propuesto.
- Entrevistas: se realizan entrevistas al cliente para adquirir conocimientos acerca del proceso de gestión de cursos como aporte significativo a la investigación e identificar las necesidades reales. A través de este método se recopila información acerca de cómo debía comportarse el sistema

El contenido del documento está estructurado por tres capítulos:

Capítulo 1: Fundamentación teórica acerca del dominio conceptual y tecnológico: En este capítulo se aborda el dominio conceptual y tecnológico donde se desarrollará la aplicación y se hará alusión a conceptos, definiciones y tecnologías relacionados con el sistema propuesto. Se realizará además, un estudio de sistemas homólogos para identificar funcionalidades y tendencias de desarrollo en el mundo y establecer resultados y características comunes entre ellos.

Capítulo 2: Características de la propuesta de la solución: Detalla la propuesta de solución del sistema, realizando una descripción de los requisitos, procesos y reglas del negocio asociadas al objeto de estudio y proceso de automatización del sistema. Se da a conocer la propuesta del sistema y los diagramas para comprender el funcionamiento del mismo.

Capítulo 3: Implementación y prueba del sistema: Se realizan pruebas funcionales al sistema para avanzar a la fase de validación y obtener los resultados del funcionamiento del mismo para verificar que cumple con el objetivo por el cual se concibió el desarrollo.

Capítulo 1: Fundamentación teórica acerca del dominio conceptual y tecnológico.

1.1 Introducción

En el presente capítulo se definen los principales conceptos y definiciones relacionados con la investigación. Además, se conceptualizan y describen los entes primordiales que rigen el acceso abierto, sus principales manifestaciones y perspectivas para Cuba y el resto del mundo. Se realiza un estudio acerca de las principales iniciativas de trabajo colaborativo enmarcadas dentro del Movimiento de Acceso Abierto, características y objetivos que persiguen. En último lugar, se hace alusión al uso de tecnologías, metodologías y herramientas de apoyo que intervienen en el proceso de construcción del producto final.

1.2 Definiciones importantes.

Acceso abierto

En el mundo los científicos que pertenecen a organizaciones estatales se enfrentan a la limitación de no poder pagar los altos costos que involucra la publicación tradicional. En consecuencia, no pueden acceder muchas veces a sus propios trabajos a pesar de que son ellos mismos los que sustentan el contenido de las revistas científicas. Por tanto, ofrecen gratuitamente su trabajo en busca de prestigio académico en base al reconocimiento de sus publicaciones.

En aras de eliminar estas barreras económicas, legales y tecnológicas y tratando de obtener a cambio una mayor accesibilidad para los documentos y una superior visibilidad para los autores surge el Movimiento de Acceso Abierto.

El inicio del Movimiento de Acceso Abierto, tiene sus orígenes varias décadas atrás, pero no es sino hasta la década de los 90 cuando toma fuerza debido a la era digital, donde la información se vuelve mucho más accesible para todos y las herramientas digitales permiten una distribución más democrática de las investigaciones académicas gracias a Internet.

El acceso abierto (en inglés *Open Access*) es un movimiento que promueve el acceso libre y gratuito a la literatura científica, fomentando su libre disponibilidad en Internet y permitiendo a cualquier usuario su lectura, descarga, copia, impresión, distribución o cualquier otro uso legal de la misma, sin ninguna barrera financiera, técnica o de cualquier tipo. La única restricción sobre la distribución y reproducción es dar al autor el control sobre la integridad de su trabajo y el derecho a ser adecuadamente reconocido y citado (SALAMANCA, 2014).

Entre los avances que promueve este movimiento se encuentran:

- Permite que los documentos estén disponibles libremente, sean más consultados y tengan más posibilidades de ser citados.
- Los conocimientos y avances científico-técnicos se distribuyen de la manera más amplia posible.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

- Devolver a la sociedad los beneficios de las inversiones realizadas en investigaciones científicas.

Con los avances del movimiento internacional de acceso abierto y el uso de repositorios digitales multidisciplinares, repositorios institucionales y bibliotecas digitales, las perspectivas de construir capacidad científica en países en desarrollo han mejorado significativamente.

El Movimiento de Acceso Abierto en Iberoamérica realiza actualmente un conjunto de iniciativas para el desarrollo de repositorios institucionales. Recientemente, en el marco de la Red CLARA (Cooperación Latino Americana de Redes de Avanzada), el Banco Interamericano de Desarrollo (BID) ha brindado apoyo para la conformación de una Red Federada de Repositorios Institucionales de Documentación Científica en América Latina. Por otra parte, Latindex ha iniciado un Portal de Portales de Revistas de América Latina y el Caribe. Son ejemplos de nuevos programas regionales que buscan articular iniciativas nacionales y regionales para avanzar en el acceso abierto a la producción científica del conjunto de países de América Latina y el Caribe, y su interoperabilidad a nivel iberoamericano e internacional (MÉXICO, 2013).

A estas alturas del movimiento más de diez mil docentes universitarios se han unido a favor del acceso abierto y recientemente la Universidad de Harvard, una de las más prestigiosas del mundo, se ha abierto al debate sobre la democratización del acceso a la información debido a los altos costos que las editoriales representan para la universidad al acceder a sus artículos. La biblioteca de dicha institución pagó durante los últimos seis años, cerca de tres millones de dólares al año, un pago insostenible; por ello, además de abrir el debate, han solicitado a su personal académico que envíen sus trabajos a revistas de acceso libre, juntando firmas de dos mil cien investigadores (GIL, 2013).

Existen dos categorías específicas de Acceso abierto: las revistas científicas y los repositorios institucionales.

Una **revista científica** se define como una publicación periódica que publica artículos científicos y/o información de actualidad sobre investigación y desarrollo acerca de un campo científico determinado (CSIC, 2009).

Los **repositorios institucionales** de acuerdo con la definición de Clifford Lynch (2003) son un conjunto de servicios ofrecidos a la comunidad científica para la gestión y disseminación de los materiales en forma digital generados por su comunidad.

1.3 Estado actual y perspectivas del Movimiento de Acceso Abierto.

El número de revistas científicas activas editadas en el mundo supera los 93.000 títulos (de los cuales 54.358 poseen revisión externa), unas cifras nada desdeñables que se han ido incrementando en los últimos años. Dentro de este universo, el número de revistas en acceso abierto supera los 11.000 títulos,

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

disponiendo de revisión externa 6.798 de ellos. La Figura 1 muestra estas cifras, así como la relación entre ellas (COMUNICACIÓN, 2012).

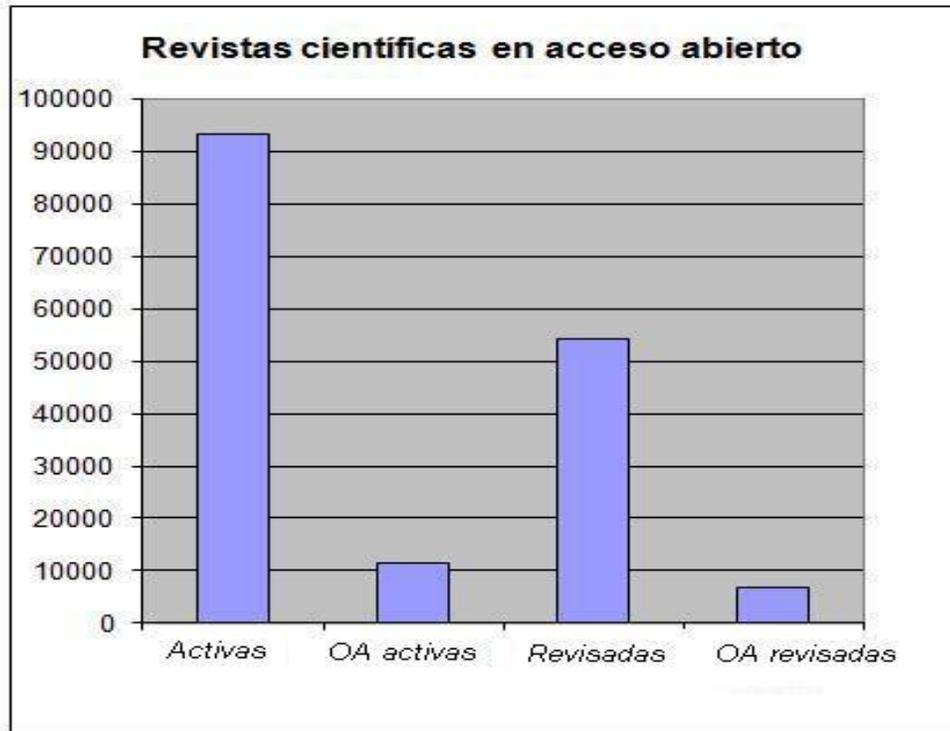


Figura 1: Revistas científicas en acceso abierto.

El Movimiento de Acceso Abierto plantea un escenario de cambios en los modelos de publicación y difusión de los conocimientos que genera la ciencia, procurando reducir las barreras de acceso y contribuyendo a incrementar la visibilidad y la preservación a largo plazo de la producción científica de los países e instituciones.

En medio de las controversias entre quienes se proclaman a favor o en contra del acceso abierto, hay muchos intereses económicos, políticos e ideológicos que son ajenos y están muy lejos de los fines beneficiosos de contribuir a la democratización de los conocimientos científicos que persigue la comunidad científica.

OpenCourseWare

Las universidades son el repositorio del conocimiento humano y el núcleo principal de expansión del conocimiento científico. Por tanto, si las instituciones publican los contenidos de los cursos que se dictan a sus estudiantes, en páginas web públicas, se contribuye a la materialización de la educación del siglo XXI, una educación a la cual puedan acceder todas las personas del mundo, y se permite intercambiar información de calidad entre estudiantes y profesores.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

También han irrumpido recientemente dentro del panorama educativo los *OpenCourseWare* cambiando en gran medida los paradigmas tradicionales en los que la educación se limitaba a los materiales disponibles.

OpenCourseWare es actualmente uno de los repositorios de conocimiento más difundido. Según *OCW Consortium*, OCW es una publicación digital libre y abierta de alta calidad de nivel universitario, estos materiales están organizados como cursos, y frecuentemente incluyen materiales de planificación de cursos y herramientas de evaluación, así como contenido temático (VALKENBURG, 2011).

El modelo para compartir conocimiento en abierto que tomó el nombre de *OpenCourseWare* fue propuesto por el *Massachusetts Institute of Technology* (MIT), quien fue el primero en implantarlo en su propia institución. Nació en el año 2000 y fue la estrategia recomendada por un comité de expertos ante el creciente impacto de Internet en la Educación Superior y, en particular, como alternativa al entonces emergente campo de la educación a distancia. Siempre de acuerdo con la misión del Instituto de diseminar conocimiento, en lugar de diseñar nuevos programas para su impartición a distancia, se ponían a disposición de los usuarios de Internet los contenidos docentes que se utilizaban en las clases del MIT. OCW persigue como objetivo generar beneficios globales frente a beneficios propios. Docentes de cualquier parte del mundo podían utilizar esos materiales en la creación de los suyos propios y de todo el globo se podían beneficiar de dichos recursos.

Actualmente existen más de 2000 cursos con acceso libre en este repositorio, en la Figura 2, se plasman los tipos de usuarios y su nivel de uso del proyecto OCW.

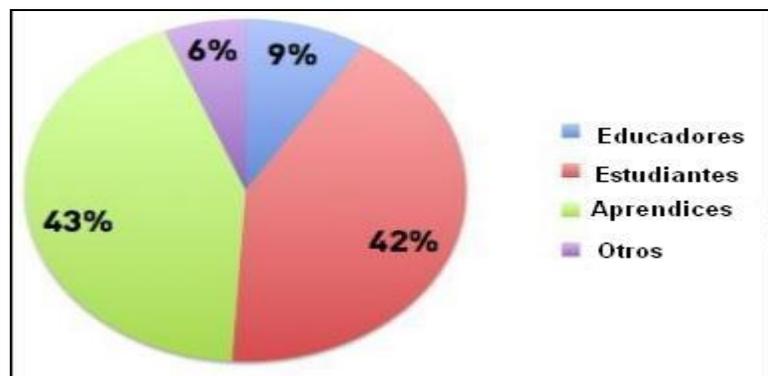


Figura 2: Usuarios de OCW.

1.5 Iniciativas de trabajo colaborativo en el mundo.

Numerosas organizaciones a nivel mundial se han dado a la tarea de impulsar los repositorios con la producción de la investigación que generan sus universidades e investigadores, sumándose a diferentes iniciativas de trabajo colaborativo que han surgido alrededor del mundo y que plantean soluciones aplicables a situaciones específicas. Algunos de los más difundidos son: *Dawning of the Dutch Network of*

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

DARE, Red de Bibliotecas Virtuales de Ciencias Sociales de América Latina y el Caribe CLACSO, Proyecto Sherpa y El protocolo OAI-PMH.

Dawning of the Dutch Network of DARE

DARE es una iniciativa común de universidades holandesas cuyo objetivo es hacer que su producción académica sea digitalmente accesible. Entre sus metas principales se destacan:

- la implementación de la infraestructura básica para el funcionamiento e interacción de repositorios
- estimular el desarrollo de servicios basados en la información que se obtiene a partir de las investigaciones que están disponibles a través de la infraestructura mencionada en el punto anterior
- iniciar y promover el envío de contenido científico hacia los repositorios y el uso del mismo (KUIL, 2004).

Red de Bibliotecas Virtuales de Ciencias Sociales de América Latina y el Caribe

La Red de Bibliotecas Virtuales de Ciencias Sociales de América Latina y el Caribe (CLACSO), a la que se puede tener acceso por Internet y sin ningún costo, tiene como objetivo principal el promover y facilitar el acceso a los resultados de las investigaciones de los centros que son miembros. Entre los servicios que proporciona esta Red de Bibliotecas sobresalen la Sala de Lectura con textos completos de libros, artículos, ponencias y documentos de trabajo publicados por la red CLACSO y otras instituciones; Bases de datos sobre la producción académica de los centros miembros: registros bibliográficos de las publicaciones, investigaciones con descripción de cada investigación, investigadores con dirección de correo electrónico de contacto; enlaces a bibliotecas y bases de datos de ciencias sociales (SOCIALES, 2010).

Proyecto Sherpa

Se dedicaba a investigar cuestiones de futuro en cuanto a comunicación académica y a desarrollar Repositorios Institucionales (RI) *Open Access* en varias universidades para facilitar la difusión mundial rápida y eficiente de la investigación.

Entre las metas de este proyecto resaltan:

- Poner en marcha RI *Open Access* de *eprints* que utilicen el protocolo OAI-PMH y que utilicen el *software Eprints*.
- Investigar cuestiones claves en la creación, población y mantenimiento de colecciones de *eprints*, incluyendo: propiedad intelectual, control de calidad, políticas de desarrollo de colecciones, modelos de negocios, cultura de comunicación académica y estrategias institucionales.
- Trabajar con los proveedores de servicios *Open Archives Initiative* para alcanzar estándares

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

aceptables (técnicos, en metadatos y de manejo de colecciones) para la difusión eficaz del contenido.

- Investigar la preservación digital de *eprints* usando el modelo de referencia *Open Archival Information System* (OAIS).
- Divulgar sus experiencias y proporcionar consejos a otros que desean instalar servicios similares.

El proyecto finalizó en enero de 2006, pero mucho de su trabajo en defensa y ayuda en el establecimiento de RI está continuando con el proyecto SHERPA Plus.

Estándares de metadatos

Los metadatos constituyen información estructurada que describe a otra información y que nos permite encontrarla, gestionarla, controlarla, entenderla y preservarla en el tiempo. En este sentido, un estándar establece un sistema común de terminologías y definiciones para documentar datos, normando la forma de describirlos. Es casi imprescindible que las estructuras y definiciones de metadatos deban tener su referencia en un estándar.

Esquema de metadatos *Dublin Core*

Dublin Core es probablemente el estándar de metadatos más extendido. Los repositorios institucionales utilizan este esquema de metadatos para describir el contenido de sus objetos.

La iniciativa de metadatos *Dublin Core* es una organización dedicada a la promoción y difusión de normas interoperables sobre metadatos y el desarrollo de vocabularios especializados en metadatos para la descripción de recursos que permitan sistemas de recuperación más inteligentes. El resultado de la primera reunión de 1995 fue un conjunto de elementos de metadatos que facilitan esta labor. Actualmente es la iniciativa más extendida en Internet y es considerada un estándar (ISO-15836-2003). El *Dublin Core* se destaca por su deliberada sencillez. Pretende ofrecer a los creadores de materiales digitales en general una lista de descriptores sencillos, fáciles de entender y que no requieran de amplios conocimientos en catalogación para utilizarlas.

El conjunto de elementos *Dublin Core* se centró en 13 elementos, pero concluyó con 15 descriptores como resultado de un consenso y un esfuerzo interdisciplinar e internacional. Ya existen transcripciones a 20 idiomas y ha sido adoptado por el CEN/ISS (*European Committee for Standardization / Information Society Standardization System*) y posee dos RFCs de Internet (*Requests for Comments*) (RFC2413) y (RFC2731). Es también el estándar oficial del *WWW Consortium* y el estándar del Z39.50. Los metadatos *Dublin Core* han sido aprobados por el organismo nacional de estandarización norteamericano (ANSI/NISO Z39.85) y los utilizan como base tanto gobiernos como agencias supranacionales y muchas otras iniciativas de metadatos pertenecientes a comunidades específicas como bibliotecas, archivos, en educación, negocios, etc. (LAPUENTE, 2013).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

El conjunto de elementos de metadatos *Dublin Core* es un conjunto de metadatos previsto para describir documentos. Los elementos poseen etiquetas descriptivas que pretenden transmitir un significado semántico a los mismos. Podemos clasificar el conjunto de elementos *Dublin Core* en 3 grupos que indican la clase o el ámbito de la información que contienen:

Contenido del curso	Recurso(como propiedad intelectual)	Instanciación del recurso
<i>Title</i> (titulo)	<i>Creator</i> (autor)	<i>Date</i> (fecha)
<i>Subject</i> (tema)	<i>Publisher</i> (editor) y otras colaboraciones	<i>Type</i> (tipo de recurso)
<i>Description</i> (descripción)	<i>Contributor</i> (otros autores/colaboradores)	<i>Format</i> (formato)
<i>Source</i> (fuente)	<i>Rights</i> (derechos)	<i>Identifier</i> (identificador)
<i>Language</i> (lenguaje)		
<i>Relation</i> (relación)		
<i>Coverage</i> (cobertura).		

Tabla 1: Elementos *Dublin Core*.

Protocolo OAI-PMH (*Open Archives Initiative - Protocol Metadata Harvesting*)

El protocolo para la transmisión de contenidos en Internet denominado OAI-PMH (*Open Archives Initiative –Protocol for Metadata Harvesting*) es el estándar utilizado para facilitar la disponibilidad de cualquier tipo de documento digital procedente de distintos repositorios, por medio de la transmisión de metadatos a través de la *World Wide Web*, utilizando estándares abiertos HTTP (*Hypertext Transport Protocol*) y XML (*eXtensible Markup Language*). Su premisa fundamental es que ofrece un marco para la interoperabilidad de repositorios independiente de la plataforma o aplicación que utilicen, mediante la codificación de sus metadatos en el esquema *Dublin Core*. La simplicidad en el uso e implementación de este esquema soluciona los problemas derivados de las conversaciones entre múltiples formatos al convertir sus datos a un formato común.

Características esenciales del protocolo OAI – PMH:

- Simplicidad: Se concibió bajo la premisa de la sencillez. Conscientes de los problemas de implementación que habían tenido otras iniciativas anteriores, los creadores buscaron una fórmula simple que estuviera al alcance de cualquier potencial implementador.
- Normalización: Basado en estándares ampliamente utilizados en Internet como son el protocolo HTTP para la transmisión de datos y órdenes y XML para la codificación de los metadatos.

- **Recolección:** Frente a otros sistemas de agregación de contenidos como los de búsqueda distribuida (Z39.50) o los sistemas de sindicación de contenidos vía RSS (de las siglas en inglés *Really Simple Syndication*), OAI-PMH ha optado por la recolección de metadatos. En este caso, existe una entidad que pone a disposición de los interesados información bibliográfica sobre los documentos que almacena. Estos, normalmente agregadores de contenidos, recogen periódica y sistemáticamente todos o parte de los metadatos expuestos para, localmente, implementar servicios de valor añadido (GUZMÁN, 2006).

1.6 *OpenCourseWare* en el mundo. Análisis de sistemas homólogos.

Los países abanderados en la implantación de sistemas OCW son: Estados Unidos, Reino Unido y otros países europeos. En el caso del mundo de habla hispana, España es el país más adelantado, esta situación contrasta con la existente en América Latina, donde el crecimiento de los OCW es pobre. En el caso de Cuba, solo el sector de la salud, como ha sucedido en casi todos los temas de acceso abierto, tiene algunos resultados. Lo cual es contradictorio puesto que a pesar de que los resultados de la educación de Cuba son reconocidos a nivel internacional, este tipo de sistema complementario al proceso formativo y que fomenta el uso de las Tecnologías de la Información y las Comunicaciones no se explota en su magnitud.

OpenCourseWare de la Universidad Carlos III de Madrid (UC3M)

La Universidad Carlos III de Madrid promueve la difusión en abierto de material de apoyo a la formación, incentivando la participación de su profesorado en dicha iniciativa. Por ello, la UC3M se adhiere al proyecto del MIT, que ha generado la agrupación de universidades de alto prestigio internacional en torno al Consorcio *OpenCourseWare*. La misión de esta acción es promocionar la educación y potenciar el conocimiento de manera abierta y sin restricciones.

Este sitio se basa en *eduCommons 4.1* que es un *software* diseñado específicamente para gestionar la publicación de contenidos en proyectos *OpenCourseWare*, como el *OpenCourseWare* de la Universidad del Estado de Utah.

Incluye una completa gestión del flujo de trabajo que ordena la organización en categorías y cursos, el almacenamiento, la modificación, la introducción de *metadata*, el control de calidad, la declaración de *copyright* y, finalmente, la publicación de los materiales docentes.

Implementado en *Python*, *Zope* y *Plone*, *eduCommons* es un programa de *software* libre que funciona sobre plataformas Linux, Mac OS X y Windows. Se distribuye bajo licencia GNU *General Public License*.

Usando las licencias *Creative Commons*, los profesores especifican las condiciones de uso que darán a los contenidos para que se respeten sus derechos de propiedad. La aceptación de esta licencia conlleva a los siguientes derechos y deberes:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

➤ Derechos:

- Se cede a los usuarios el derecho de usar y distribuir los materiales en su formato original o de forma adaptada.
- Se permite a los usuarios crear trabajos derivados mediante su edición, traducción, añadiéndolos, combinándolos o incorporándolos a otros materiales de otros autores.

➤ Deberes:

- Se exige que los materiales utilizados sean atribuidos a la institución y a los autores originales.
- Se restringe el uso a propósitos educativos no comerciales.
- Se requiere que si el original o los materiales derivados se publiquen o distribuyan de nuevo, sean ofrecidos libremente a otros usuarios bajo idénticas condiciones de licencia de uso (MADRID, 2014).

OpenCourseWare de la Universidad Politécnica de Cartagena (UPCT)

La Universidad Politécnica de Cartagena también se suma a las iniciativas de compartir información y en 2008 firma su adhesión al consorcio OCW Universia creando su portal OCW UPCT.

Dentro del portal, los cursos se han estructurado por Centros. Parten del principio de que OCW no es una plataforma de enseñanza virtual y que únicamente proporciona acceso libre al material docente de las asignaturas, no proporcionando contacto alguno con profesores. OCW tampoco otorga certificados ni títulos.

➤ Estructura de las asignaturas de OCW UPCT:

Los contenidos de las diferentes asignaturas se ofrecen siguiendo una misma estructura con el fin de ofrecer contenidos homogéneos: presentación, programa, material de clase, prácticas y ejercicios, exámenes, otros recursos y bibliografía.

➤ Ventajas de publicar en OCW UPCT:

- Facilita la visibilidad del trabajo docente del profesor.
- Enriquece la labor docente al tener acceso a las asignaturas similares de diversas universidades.
- Ofrece la posibilidad de compartir material educativo con profesores de asignaturas similares en otras universidades.
- Motiva la mejora de la calidad de las asignaturas al someter su contenido a la revisión pública.

➤ Propiedad intelectual del material del OCW UPCT

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

- No poner ningún contenido como propio del que no seamos propietarios. En caso de utilizar material del que no seamos propietarios es obligatorio citar la autoría.
- Las licencias que utilizamos son las que ofrece *Creative Commons* que facilitan la gestión de los derechos de autor.

➤ *Licencias Creative Commons* utilizadas para los materiales docentes del OCW UPCT

A los materiales docentes se les asigna una licencia Reconocimiento – No comercial – Compartir bajo la misma licencia, lo que quiere decir que:

- El material docente puede ser utilizado por terceras personas siempre que se especifique y cite a su autor.
- El material docente no puede ser usado para fines comerciales.
- El material docente generado a partir del mismo, sólo puede distribuirse bajo una licencia idéntica a esta (MOLINA, 2013).

OpenCourseWare Universia

Esta iniciativa ha tenido un gran impacto y ha impulsado a que otras universidades de prestigio internacional se hayan adherido, generando proyectos propios en coordinación con OCW-MIT. Este el caso de *OpenCourseWare* Universia que, a través de medios digitales y en abierto, difunde los recursos docentes estructurados en asignaturas. En definitiva se concibe como una muestra para la difusión internacional de las actividades de nuestras universidades.

Fundación Universia se ha comprometido con este proyecto estableciendo un acuerdo de colaboración con la Universidad Politécnica de Madrid para impulsar conjuntamente su desarrollo.

Este proyecto cuenta igualmente con el apoyo de Universia, que promueve su desarrollo y expansión a través de sus portales universitarios y de las universidades de su red (UNIVERSIA, 2014).

OpenCourseWare de la Universidad Nacional de Córdoba (UNC):

Este OCW proporciona el acceso libre y sencillo a obras intelectuales generadas por profesores para cursos en la UNC, difundiéndolas y simultáneamente protegiendo su autoría. Contribuye a la difusión y construcción pública del conocimiento universal y con ellas, al bien público, confluyendo con el movimiento mundial por el acceso abierto a obras intelectuales, en este caso pedagógicas y de nivel universitario. Difunde además los frutos del trabajo pedagógico de los profesores de la UNC, dándoles visibilidad internacional. Colabora en la consolidación de un modelo basado en estándares que pueda ser compartido por otras universidades (CÓRDOBA, 2013).

OpenCourseWare de la Universidad de León (ULE):

Se trata de un espacio web que contiene materiales docentes creados por profesores para la formación superior:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

- Los materiales representan un conjunto de recursos (documentos, programa, calendario,..) utilizados en el proceso de enseñanza-aprendizaje de las asignaturas que se imparten en las aulas de la ULE.
- Los materiales se ofrecen libremente y son accesibles universalmente vía web.
- Su difusión no se encuentra limitada por temas relativos a la propiedad intelectual y derechos de *copyrights*. Todos los materiales están publicados bajo licencia *Creative Commons*.
- Se permite el uso, la reutilización, la adaptación y la distribución por terceros siempre que sea sin fines comerciales, sean atribuidos a la institución que lo publica originalmente y si procede al autor y el material resultante tras su uso debe ser de libre utilización por terceros y se encuentra sujeto a estos mismos requisitos.

La ULE ha asumido el compromiso de “compartir el conocimiento” promoviendo que los profesores que lo deseen “hagan visibles” los materiales docentes utilizados para impartir sus asignaturas, con ello se ofrece a otros profesores, estudiantes o profesionales de cualquier lugar del mundo un acceso abierto y libre al conocimiento (LEÓN, 2012).

Resultado del estudio de sistemas homólogos.

OpenCourseWare es un espacio web que contiene materiales docentes creados por profesores para la formación superior. Las características que distinguen al proyecto *OpenCourseWare* de iniciativas similares son las siguientes:

- Los recursos didácticos publicados en un sitio web OCW se organizan por ramas de conocimientos y en unidades de “asignaturas” o “cursos”. Con ello se quiere indicar:
 - Los accesos se realizan por rama de enseñanza/asignatura e incluyen un conjunto significativo de todas las enseñanzas recogidas en dicha asignatura.
 - Los materiales se ofrecen de forma organizada por categorías: programa de la asignatura, lecturas obligatorias, materiales de clase, ejercicios, guía de aprendizaje.
- Los autores garantizan que el material que publican en el sitio web OCW es original o tienen los derechos, bien directamente por ser propietario o bien a través del tipo de licencia que los soporta, para ser reutilizados en “abierto” sin infringir los “*copyright*” de otras personas o instituciones.
- Son accesibles universalmente a través de la red:
 - Sin limitaciones geográficas.
 - Sin exclusión de usuario, ni necesidad de registrarse o utilizar palabras claves de acceso.
 - No exigen requisitos técnicos más allá de un navegador web, un lector universal y gratuito de documentos.

- Los sitios web OCW no ofrecen a los usuarios un servicio de educación a distancia y por tanto no llevan asociado la asistencia por y/o disponibilidad los profesores/autores de los mismos. Un sitio OCW no dispondrá por tanto de foros, correo u otros medios de interacción entre el profesor y el alumno. Tampoco se podrá reclamar cualquier acreditación o reconocimiento por parte de la institución que publique dichos contenidos.

1.7 Descripción del entorno tecnológico para la creación de un *OpenCourseWare*.

1.7.1 Sistemas de Gestión de Contenidos.

Para comprender mejor el análisis y uso del Sistema de Gestión de Contenidos Drupal así como los diferentes lenguajes con los que trabaja, se precisan un conjunto de definiciones que tienen estrecha relación con estos sistemas.

Un Sistema de Gestión de Contenidos-CMS (*Content Management System*, en inglés) es un *software* que permite la creación y administración de los contenidos de una página web, principalmente, de forma automática. Así, con él se puede publicar, editar, borrar, otorgar permisos de acceso o establecer los módulos visibles para el visitante final de la página (MAYOR, 2013).

Los CMS son aplicaciones altamente adaptables que brindan la posibilidad de manejar contenidos de propósito general, aunque se pueden personalizar todo lo que se desee. A partir de estas definiciones se puede concluir que su principal objetivo es proveer al desarrollador de una herramienta para la construcción de aplicaciones web que operen contenidos de forma dinámica y reduzcan la necesidad de conocimientos técnicos en cuanto a programación se refiere.

EL uso de CMS facilita bastante las tareas de mantenimiento de las páginas web, debido a que en ellos está separada la presentación del contenido lo que permite modificar los ficheros de presentación para cambiar el aspecto, e incluso la arquitectura del sitio sin perder los contenidos y el acceso a los mismos. Se asume como contenido todo lo que pueda generarse desde la aplicación: noticias, artículos, efemérides, entre otros, que pueda almacenarse en la base de datos u otro lugar.

Sistemas de Gestión de Contenidos más utilizados en la actualidad.

Los CMS se pueden clasificar en cuanto a lo que a licencia se refiere, en desarrollados y comercializados por empresas, y por otra parte, los de código abierto, que son aquellos que permiten que se le realicen cambios al producto, y el acceso libre a este, y por tanto, posibilita el desarrollo de nuevas funciones. Debido a los altos costes que implica el soporte profesional que brindan los CMS desarrollados por empresas se decide para la presente investigación la utilización de un CMS de código abierto. Estos últimos generalmente comparten la información necesaria y ayuda en la solución de diversos problemas y existen algunas empresas que ofrecen servicios de valor añadido a los usuarios que los utilizan.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

Existe una variedad de CMS de código abierto extraordinariamente amplia y cada uno de ellos posee las funcionalidades generales de cualquier gestor de contenidos, además de características específicas que los identifican. Por consiguiente, para elegir uno de ellos se debe lograr una correlación entre los siguientes elementos: dominio en cuanto a funcionalidad, buen funcionamiento de su sistema de seguridad, y simpleza en cuanto a su uso. Entre los CMS más difundidos se encuentran Plone y Drupal.

Plone: Es un CMS muy flexible y poderoso, que presenta una excelente interfaz de usuario, una instalación muy limpia, un impresionante grado de personalización, y una integración con *Lightweight Directory Access Protocol* (LDAP) u otros sistemas de *login*. “Posibilita además construir un sitio cuyo contenido se puede enriquecer constantemente, debido a sus facilidades de actualización e interacción, y con ello, obtener ventajas competitivas. Pero, probablemente su característica más importante es que es un sistema de código abierto y libre.

Entre sus características principales se encuentran:

- Puede utilizarse como servidor Intranet o Extranet, Sistema de Publicación de documentos y una herramienta de trabajo en grupo para colaborar entre entidades distantes.
- Es completamente modularizado o sea, los módulos se pueden gestionar.
- Permite la personalización extensible para los principales elementos de seguridad.
- Permite la gestión de usuarios y además tiene un sistema de permisos de grupo de gran alcance y de uso fácil.

La principal desventaja de Plone radica en la necesidad de emplear especialistas para el desarrollo de interfaces de usuario, diseñadores gráficos que mejoren la apariencia y la cascada de estilo que este sistema posee.

Drupal: es un sistema de administración de contenido para sitios web. Permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, *Blogging* y administración de usuarios y permisos basados en PHP. Drupal es un sistema dinámico, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno web incluido en el producto (FOUNDATION, 2012).

Entre las características generales de Drupal se encuentran:

- Posee un robusto sistema de ayuda *online* y páginas de ayuda, tanto para usuarios como para administradores.
- Todo el contenido de Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.

- El código fuente está libremente disponible bajo los términos de la licencia pública general de GNU (GNU/GPL). Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- Aunque la mayor parte de las instalaciones de Drupal utilizan el sistema de gestión de base de datos *MySQL*, existen otras opciones. Drupal incorpora una capa de abstracción de base de datos que actualmente está implementada y mantenida para *MySQL* y *PostgreSQL* que constituye otro sistema de gestión de base de datos muy poderoso, aunque permite incorporar fácilmente soporte para otras bases de datos.
- Drupal ha sido diseñado desde el principio para ser multiplataforma. Puede funcionar con *Apache* o *Microsoft IIS* como servidor web y en sistemas como *Linux*, *Solaris*, *Windows* y *Mac OS X*. Por otro lado, al estar implementado en *PHP*, es totalmente portable.

Selección del Sistema de Gestión de Contenidos.

Drupal es en la actualidad un líder destacado en el desarrollo de aplicaciones; el gran atractivo que posee se enmarca en la flexibilidad que posee, sirve tanto para crear un blog personal, como para dar cobertura a la actividad de grandes comunidades *online*. En este sentido, otro aspecto interesante de Drupal es que la interfaz de administración está integrada en la propia Web, a diferencia de otros CMS que tienen una interfaz diferente para las tareas administrativas. Las principales características que justifican la selección para el desarrollo de la aplicación del CMS Drupal en su versión 7.24 son:

- La arquitectura basada en entidades con la posibilidad de extenderla a través de campos.
- Las facilidades para la recuperación de información.
- El alto nivel de personalización.
- El alto nivel de delimitación de funciones y de implementación de flujos de trabajo.

1.7.2 Metodología de desarrollo de software.

Para apoyar el desarrollo de la solución se procede a la selección de la metodología de desarrollo que describirá el proceso de *software* de manera detallada y completa. Existen diferentes formas de comparar y/o clasificar estas metodologías debido a la diversidad de propuestas y diferencias en el grado de detalle, información disponible y alcance de cada una de ellas. Considerando su filosofía de desarrollo, aquellas metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales (o también denominadas Metodologías Pesadas, o Peso Pesado). Otras metodologías, denominadas Metodologías Ágiles, están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

De acuerdo a las siguientes características se decide utilizar una metodología ágil en el desarrollo de la propuesta de solución:

- **Incremental** (entregas pequeñas de *software*, con ciclos rápidos)
- **Cooperativo** (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación)
- **Sencillo** (el método en sí mismo es fácil de aprender y modificar, bien documentado)
- **Adaptable** (permite realizar cambios de último momento)

Dentro del grupo de metodologías ágiles se encuentran las que quedan relacionadas a continuación.

Programación Extrema (*Extreme Programming*)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. *Extreme Programming* (XP) se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Características de XP, la metodología se basa en:

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se pueda hacer pruebas de las fallas que pudieran ocurrir. Es como si nos se adelantaran a obtener los posibles errores.
- **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

Ventajas y desventajas de XP:

Una de las ventajas de la programación extrema es que se adapta al desarrollo de sistemas pequeños y grandes; optimiza el tiempo de desarrollo; permite realizar el desarrollo del sistema en parejas para complementar los conocimientos; el código es sencillo y entendible, además de la poca documentación a elaborar para el desarrollo del sistema.

Las desventajas son que no se tiene la definición del costo y el tiempo de desarrollo; el sistema crece después de cada entrega al cliente y nadie puede decir que el cliente no querrá una función más; se

necesita de la presencia constante del usuario, lo cual en la realidad es muy difícil de lograr. Otra desventaja es la programación en parejas, algunos desarrolladores son celosos del código que escriben y no les es grato que alguien más modifique las funciones que realizó o que su código sea desechado por no cubrir el estándar (VERACRUZANA, 2014).

OpenUP

OpenUP es un proceso modelo y extensible, dirigido a gestión y desarrollo de proyectos de *software* basados en un desarrollo iterativo, ágil e incremental apropiado para proyectos pequeños y de bajos recursos; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo.

OpenUP es un proceso de desarrollo de *software* mínimamente suficiente, esto quiere decir que incluye solo el contenido fundamental, esto es que no provee orientación sobre temas en los que el proyecto tiene que lidiar, como son: el tamaño del equipo, el cumplimiento, seguridad, orientación tecnológica entre otras. Sin embargo, *OpenUP* es completa en el sentido de que manifiesta por completo el proceso de construir un sistema. Para atender las necesidades que no están cubiertas en su contenido *OpenUP* es extensible a ser utilizado como base sobre la cual se pueden añadir o adaptarse a contenido de otro proceso que sea necesario (GONZÁLEZ, 2014).

OpenUP está caracterizado por cuatro principios básicos interrelacionados, a saber:

- Colaboración para unificar intereses y compartir conocimientos.
- Equilibrio de prioridades competentes a maximizar el valor de los involucrados con el resultado del proyecto.
- Enfoque en la articulación de la arquitectura.
- Desarrollo continuo para obtener retroalimentación y realizar las mejoras respectivas. *OpenUP* se centra en articular la arquitectura para facilitar la colaboración técnica, reducir el riesgo y minimizar el sobreesfuerzo de desarrollo.

OpenUP desarrolla un ciclo de vida iterativo que mitiga el riesgo a tiempo y ofrece demostrar resultados en curso al cliente del proyecto.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

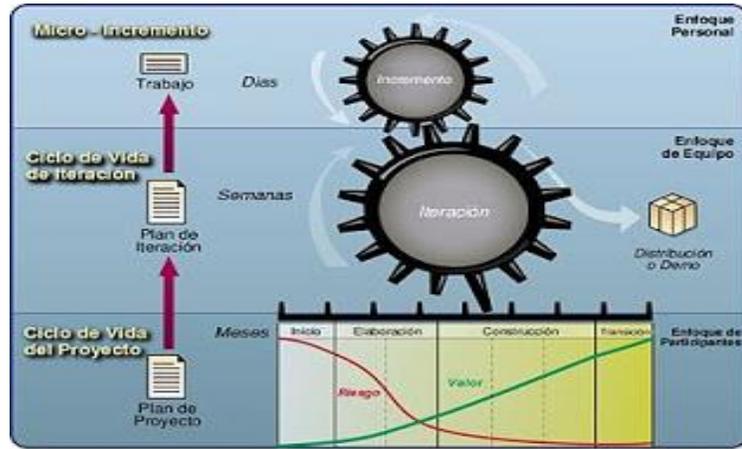


Figura 3: Ciclo de vida de OpenUP (FRAMEWORK, 2013).

Fases del OpenUP:

- **Concepción:** El propósito de esta fase es lograr concurrencia entre todas las partes interesadas sobre los objetivos del ciclo de vida del proyecto. Hay cuatro objetivos de la fase inicial, que precisan el alcance, los objetivos del proyecto y la viabilidad de la solución proyectada.
- **Elaboración:** El propósito de esta fase es establecer la línea base de la arquitectura del sistema y proporcionar una base estable para la mayor parte del esfuerzo de desarrollo en la fase siguiente. Hay objetivos para la fase de elaboración que le ayudan a afrontar los riesgos asociados con los requisitos, la arquitectura, los costos y el programa.
- **Construcción:** El propósito de esta fase es completar el desarrollo del sistema basado en la arquitectura de línea base. Hay objetivos para la fase de construcción que ayudan a tener costo-eficiente, desarrollo de un producto completo -una versión operativa de su sistema- que se puede implementar en la comunidad de usuarios.
- **Transición:** El propósito de esta fase es asegurar que el *software* está listo para su entrega a los usuarios. Hay objetivos para la fase de transición que ayudan a afinar la funcionalidad, el rendimiento y la calidad general del producto beta desde el final de la fase anterior.

Selección de la metodología de desarrollo de *software*

La selección de *OpenUP* como metodología de *software* a utilizar para el desarrollo de la solución propuesta sentó sus bases en las siguientes características:

- Metodología de desarrollo de *software* de código abierto diseñado para pequeños equipos organizados.
- Proceso iterativo e incremental que es mínimo, completo y extensible.
- Permite detectar errores tempranos a través de un ciclo iterativo.

- Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

Ventajas en el uso del *OpenUP*

- Es apropiado para proyectos pequeños y de bajos recursos permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito.
- Permite detectar errores tempranos a través de un ciclo iterativo.
- Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP.
- Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

Además, el Centro de Ideoinformática (CIDI) adopta esta metodología para el proceso de desarrollo en los diferentes proyectos productivos que pertenecen al mismo debido a que es una metodología gratis, ágil, modificable y evolutiva que se puede integrar con otras metodologías ya que pueden resolverse las tareas de desarrollo utilizando las prácticas de XP. También brinda una referencia clara y simplificada para la inducción de nuevo personal.

1.7.3 Sistema Gestor de Base de Datos

Son un conjunto de programas que permiten crear y mantener un conjunto de datos relacionados entre sí, asegurando su integridad, confidencialidad y seguridad. Posibilitan la especificación de los tipos de datos, las estructuras y sus restricciones. Su utilización permite construir una base de datos y guardarla en algún medio controlado por el mismo Sistema Gestor de Base de Datos (SGBD) donde la base de datos puede ser manipulada realizando consultas, actualizaciones y generando informes. Es una aplicación que permite a los usuarios definir, crear y mantener la base de datos proporcionando el acceso controlado a la misma. Ejemplo de SGBD son PostgreSQL y MySQL.

MySQL

Una de las características más interesantes de MySQL es que permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos. Por otro lado, MySQL es conocida por desarrollar alta velocidad en la búsqueda de datos e información, a diferencia de sistemas anteriores.

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de Interfaz de Programación de Aplicaciones (API's) en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.

6. Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

MySQL surgió como una necesidad de un grupo de personas sobre un gestor de bases de datos rápido, por lo que sus desarrolladores fueron implementando únicamente lo que precisaban, intentando hacerlo funcionar de forma óptima. Es por ello que, aunque MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas de sus principales características:

1. Subconsultas.
2. *Select INTO Table*: Esta característica propia de Oracle, todavía no está implementada.
3. *Triggers* y *Procedures*: Se tiene pensado incluir el uso de *procedures* almacenados en la base de datos, pero no el de *triggers*, ya que estos reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.
4. Transacciones: a partir de las últimas versiones ya hay soporte para transacciones, aunque no por defecto (se ha de activar un modo especial).
5. Integridad referencial: aunque sí que admite la declaración de claves ajenas en la creación tablas, internamente no las trata de forma diferente al resto de campos.

Desventajas

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (AFFILIATES, 2011).

PostgreSQL

Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema (POSTGRESQL, 2012).

Características más importantes y soportadas por PostgreSql:

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de bits, etc. También permite la creación de tipos propios.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Copias de seguridad en caliente (*Online/hot backups*).
- Regionalización por columna.
- Múltiples métodos de autenticación.
- Acceso encriptado vía SSL (de las siglas en inglés *Secure Sockets Layer*).

- Completa documentación.
- Disponible para Linux y UNIX en todas sus variantes y Windows 32/64bit.

Ventajas:

- Ampliamente popular. Ideal para tecnologías web.
- Fácil de Administrar.
- Su sintaxis SQL es estándar y fácil de aprender.
- Multiplataforma.
- Capacidades de replicación de datos.
- Soporte empresarial disponible.

Selección del Sistema Gestor de Base de Datos:

Se selecciona **PostgreSQL versión 9.1** como Sistema Gestor de Base de Datos debido a que es multiplataforma y no presenta dificultades con la integridad de los datos, lo cual es una ventaja pues no se pone en riesgo la información con la que se trabaja. Es un gestor de bases de datos seguro debido a que las transacciones que se ejecutan requieren de autenticación, de esta forma se garantiza que solo las personas autorizadas realicen esta acción. En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.

1.7.4 Lenguajes utilizados

Lenguaje de Marcado de Hipertexto (HTML) 5.0: El uso de aplicaciones web se ha enfrentado desde sus comienzos con una importante desventaja en comparación con las aplicaciones de escritorio, y era la posibilidad de estas últimas de trabajar en entornos desconectados. HTML5 aporta una importante mejora en este aspecto para permitir la utilización de todas las funcionalidades de las aplicaciones web “sin conexión” a través de API’s, que sincronizan sus datos cuando la conexión entre el dispositivo y la red se restablece.

Se decide utilizar este lenguaje a partir de que permite escribir un código conciso que ayuda a distinguir el significado del código, del estilo y contenido. Además, posibilita organizar más efectivamente los CSS (Hojas de estilo en cascada, en español). El navegador permite el uso de una base de datos local, con la que se puede trabajar en la página web por medio del cliente (WEB, 2014).

PHP (acrónimo de *Hypertext Preprocessor*) versión 5.3.8: PHP es un lenguaje “del lado del servidor” (esto significa que PHP funciona en un servidor remoto que procesa la página web antes de que sea abierta por el navegador del usuario) especialmente creado para el desarrollo de páginas web dinámicas. Puede ser incluido con facilidad dentro del código HTML, y permite una serie de funcionalidades tan

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

extraordinarias que se ha convertido en el favorito de millones de programadores en todo el mundo (VÁZQUEZ, 2003).

Para la solución propuesta se decide utilizar este lenguaje apoyado en sus características más significativas:

- **Gratuito:** Al tratarse de *software* libre, puede descargarse y utilizarse en cualquier aplicación, personal o profesional, de manera completamente libre.
- **Gran popularidad:** Existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código.
- **Enorme eficiencia:** Con escaso mantenimiento y un servidor gratuito (en nuestro caso, Apache), puede soportar sin problema millones de visitas diarias.

CSS 3.0: Se utiliza CSS debido a que es la mejor forma de separar los contenidos y su presentación además de ser imprescindible para crear páginas web complejas. La posibilidad de definir contenidos y su aspecto presenta ventajas para la presente investigación pues obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados documentos semánticos). CSS mejora además la accesibilidad, reduce la complejidad de mantenimiento y permite visualizar el documento en dispositivos diferentes. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. (ÁLVAREZ, 2010).

La utilización de CSS en el desarrollo de la solución propuesta está avalada por las siguientes características:

- Control de la presentación de la página, lo que facilita su modificación.
- Fácil creación de las plantillas al mantener la misma imagen en todas las páginas del sitio.
- Mayor accesibilidad.
- Limpieza del código fuente.
- Código HTML más legible.
- Logra que los documentos se vean igual en todos los navegadores.

JavaScript: JavaScript es un lenguaje de programación que ha permitido el gran desarrollo de la Web. Su avance más significativo radica en el logro de páginas web dinámicas y exactas en cuanto a posición y presentación de su contenido. Técnicamente es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos (EGUÍLUZ, 2009).

La solución propuesta pretende evidenciar las principales ventajas que brinda el lenguaje JavaScript entre las cuales se encuentran:

- Lenguaje de programación desarrollado para incrementar las funcionalidades del lenguaje HTML.

- Es un lenguaje orientado a eventos, se pueden desarrollar scripts que ejecuten acciones en respuesta a estos eventos.
- Sigue el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base y extendiendo su funcionalidad.

1.7.5 Lenguaje de modelado

El modelado de sistemas *software* es una técnica para tratar con la complejidad que les corresponde a estos sistemas. El uso de modelos en la presente investigación permite obtener una visualización del sistema a construir.

UML (de las siglas en inglés *Unified Modeling Language*): se decide utilizar Lenguaje Unificado de Modelado porque es un lenguaje estándar. Su utilización permite modelar sistemas de *software* en diversos dominios: sistemas de información empresariales y sistemas web. Combina lo mejor del modelado de procesos, objetos, datos y componentes. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para desarrollar un sistema de *software* previo al proceso intenso de escribir código.

1.7.6 Entornos de Desarrollo Integrados

El *Integrated Development Environment* (IDEs) o Entorno de Desarrollo Integrado, es un conjunto de herramientas para el programador, que suelen incluir en una misma suite, un buen editor de código, administrador de proyectos y archivos, enlace transparente a compiladores y *debuggers* e integración con sistemas controladores de versiones o repositorios (RAMÍREZ, 2009).

Para el desarrollo de la solución se selecciona **NetBeans** en su versión 7.2 porque permite crear aplicaciones con cualquier grado de complejidad. Soporta varios lenguajes, tales como: Java, C, C++, PHP, Ada, Groovy, PL/SQL, Python, Ruby y todos los relacionados con la web (HTML, CSS, JavaScript y XML). Se identifica por su portabilidad ya que es compatible con la mayoría de los sistemas operativos como Windows, Mac, GNU/Linux y Solaris. La plataforma admite que las aplicaciones sean perfeccionadas a partir de un conjunto de componentes de *software* llamados módulos. Estos a su vez pueden ser desarrollados independientemente, razón por la que las aplicaciones basadas en NetBeans pueden ser extendidas fácilmente por otros desarrolladores de *software* (GARCÍA, 2013).

1.7.7 Herramientas Case

Las herramientas Case (*Computer Aided Software Engineering* (en español, Ingeniería de *Software* Asistida por Computadoras)) surgen a partir de la necesidad de encontrar técnicas de trabajo que pudieran incrementar el control de la calidad en el proceso de desarrollo de *software*.

Rational Rose

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

Rational Rose es una herramienta de diseño orientada a objetos, que da soporte al modelado visual, es decir, que permite representar gráficamente el sistema, permitiendo hacer énfasis en los detalles más importantes, centrándose en los casos de uso y enfocándose hacia un *software* de mayor calidad, empleando un lenguaje estándar común que facilita la comunicación.

Es una herramienta que permite crear los diagramas que se generan durante el proceso de Ingeniería en el Desarrollo del *Software*. Permite que el equipo de desarrollo entienda mejor el problema, que identifique las necesidades del cliente en forma más efectiva y comunique la solución propuesta de forma más clara.

Rational permite completar una gran parte de las disciplinas (flujos fundamentales) de RUP tales como:

- Captura de requisitos (parcialmente).
- Análisis y diseño (completamente).
- Control de cambios y gestión de configuración (parcialmente).
- Implementación (como ayuda) (NOBREGA, 2013).

Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones.

Características principales:

- Diagramas de procesos de negocio - proceso, decisión, actor de negocio, documento.
- Ingeniería inversa. Código a modelo, código a diagrama.
- Editor de detalles de casos de uso - Entorno todo en uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas de flujo de datos.
- Generación de bases de datos - Transformación de diagramas de entidad-relación en tablas de base de datos.
- Generador de informes.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Editor de figuras (OVERVIEW, 2013).

Selección de la herramienta Case a utilizar en la propuesta de solución

Para la investigación se seleccionó como herramienta a utilizar **Visual Paradigm versión 8.0** por ser multiplataforma y permitir la operatividad con otras aplicaciones. Esta herramienta tiene entre sus particularidades soportar el ciclo completo de vida del *software*; además es fácil de utilizar y admite la integración con varios entornos de desarrollo. Así mismo, presenta soporte para aplicaciones web.

1.7.8 Servidor Web

Para el desarrollo de la solución se elige como servidor web el Apache en su versión 2.2.2. Este es un servidor web de código abierto, altamente configurable y modular. Soporta Perl, PHP y otros lenguajes *scripts*. Funciona sobre varios sistemas operativos: Linux, Solaris, Mac OS y Windows. Su función principal es analizar cualquier archivo solicitado por un navegador y mostrar resultados correctos de acuerdo con el código del archivo. Permite configurar los informes de errores, presenta visualización de códigos en numerosos niveles de HTML y la capacidad de determinar qué nivel del navegador puede aceptar el contenido. Permite la personalización de variables y tiene soporte para reparación de errores. Es uno de los primeros servidores en soportar host basados en direcciones IP (de las siglas en inglés *Internet Protocol*) y host virtuales. Tiene un elaborado índice de directorios, un directorio de alias, informe de errores HTTP configurable, gestión de recursos para procesos hijos, reescritura de las URL, comprobación de ortografía de las URL y manuales *online* (KABIR, 2010).

Apache es un servidor web que no demanda el manejo de muchos recursos y permite que muchos lenguajes sean utilizados del lado del servidor. Tiene la característica además de permitir la protección por contraseñas de las páginas de un gran número de usuarios.

Capítulo 2: Características de la propuesta de solución

2.1 Introducción

En el presente capítulo se ejemplifican y especifican las características del sistema a desarrollar. Con este fin se define el modelo de dominio según el objeto de estudio a partir del uso de la metodología *OpenUP*. Se identifican los requisitos con los que debe cumplir el *OpenCourseWare*. Se expone además la arquitectura y el diseño del sistema a desarrollar.

2.2 Propuesta del *OpenCourseWare* basado en la arquitectura Drupal.

Se pretende automatizar el flujo de trabajo del proceso editorial de un sitio *OpenCourseWare*, de manera que los profesores puedan gestionar la creación y la edición de sus cursos. Se desea también que la aplicación cuente con el proceso de evaluación y recuperación de la información. El sistema debe ser capaz de automatizar procesos como la gestión de categorías y el envío de boletines a usuarios interesados en acceder a cursos pertenecientes a una categoría determinada.

Los principales objetos de información con los cuales se opera en el sistema son referentes a artículos, cursos, materiales, profesores y categorías. Los materiales son agrupados dentro de los cursos que a su vez corresponden a una categoría. Estas acciones las realiza un usuario con los permisos necesarios. Se gestionan los roles de usuario y sus permisos para gestionar las acciones que se realizan en el sistema. Los roles principales que intervienen en el proceso son: el profesor, el editor y el administrador.

Un profesor tendrá la opción de agregar cursos al sistema precisando un conjunto de datos como nombre, categoría a la que pertenecerá, imagen que no debe exceder los 2 megabytes, la descripción del mismo entre otros campos. Una vez creado el curso pasa a la lista de cursos por aprobar, el editor será quien gestione la lista de cursos incluyéndolo en la lista de cursos aprobados o pendientes.

Otra información a gestionar son los artículos que se podrán publicar en el sitio. El editor es el encargado de realizar estas publicaciones. El administrador por su parte gestionará las categorías y las notificaciones que serán enviadas a los usuarios interesados en una categoría determinada. La gestión de servicios OAI-PMH se realiza a través de la generación de un archivo XML ante una petición HTTP que se le realiza al sistema.

2.3 Modelo de dominio.

El modelo de dominio es una representación visual estática del entorno real objeto del proyecto. Se dice que es estática porque no representa la interacción en el tiempo de los objetos, sino que representa una visión "parada" de las clases y sus interacciones. En él se representan los conceptos más importantes y significativos en el desarrollo de un sistema. Su objetivo fundamental es definir las interrelaciones de los

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

objetos más importantes representados mediante clases. Además, desempeña un papel central en la comprensión del entorno actual y en la planificación futura de la posible aplicación.

Captura los elementos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los entes que existen o los eventos que acontecen en el entorno en el que trabaja el sistema. Muchos de los objetos o clases pueden adquirirse de una especificación de requisitos o mediante la entrevista con los expertos del dominio (LARMAN, 2003).

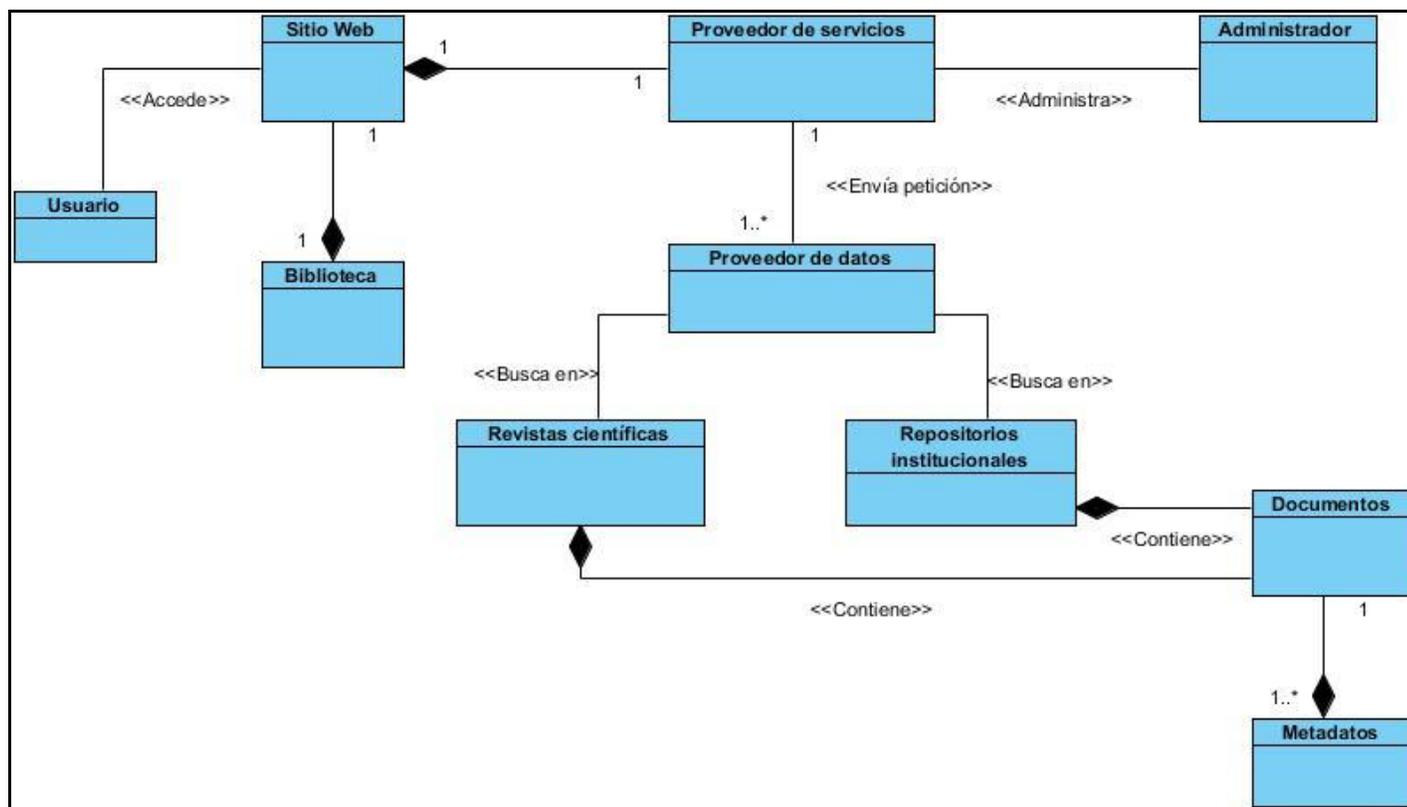


Figura 4: Diagrama del modelo de dominio.

2.3.1 Descripción de las clases conceptuales y roles del dominio.

A continuación se describen los diferentes objetos y relaciones que participan en el proceso.

Administrador: persona que cuenta con permisos de administración sobre el sitio web.

Usuario: persona que accede al sistema.

Proveedor de servicios: herramienta que se encarga de efectuar peticiones al proveedor de datos para recuperar información bibliográfica y procesarla en el sistema.

Proveedor de datos: mecanismo que se encarga de la recolección de metadatos a partir de las peticiones generadas por el proveedor de servicios.

Metadatos: información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato u objeto que tiene la finalidad de facilitar su recuperación.

Revista Científica: publicación periódica de la cual se escogen artículos científicos y/o información de actualidad sobre investigación y desarrollo para ser publicados en el sitio.

Repositorio: archivos o depósitos en sitios centralizados donde se almacena y mantiene la información digital que será consumida por el proveedor de servicios.

Biblioteca: entidad que hace uso de la aplicación para exponer y recuperar contenidos.

Sitio Web: aplicación publicada en la gran red de redes de la biblioteca que gestiona los contenidos.

2.4 Levantamiento de requisitos.

Para el éxito del desarrollo del portal fue necesario tener en cuenta las características y cualidades que el mismo debe tener, estas se agrupan en requisitos que debe cumplir el producto y que parten de la buena comunicación entre la empresa y el equipo de desarrollo. “Existen dos tipos de requisitos, los funcionales que son las condiciones que un producto debe cumplir y los no funcionales que representan las cualidades que el producto debe tener” (CARO, 2012).

2.4.1 Requisitos funcionales.

RF 1: Autenticar usuario

RF 2: Contactar administrador

Gestionar cursos

RF 3: Agregar cursos

RF 4: Eliminar cursos

RF 5: Editar cursos

Gestionar materiales

RF 6: Crear materiales

RF 7: Editar materiales

RF 8: Eliminar materiales

RF 9: Mostrar materiales

Gestionar usuarios

RF 10: Crear usuario

RF 11: Modificar usuario

RF 12: Mostrar usuario

RF 13: Eliminar usuario

Gestionar categorías

RF 14: Gestionar campos de categoría

RF 15: Editar categoría

RF 16: Listar categorías

RF 17: Gestionar presentación

Gestionar lista de cursos

RF 18: Adicionar un curso a la lista de cursos

RF 19: Modificar lista de cursos

RF 20: Eliminar lista de cursos

RF 21: Mostrar lista de cursos

Gestionar servicios OAI-PMH

RF 22: Generar respuestas XML a peticiones OAI-PMH

Gestionar boletines

RF 23: Crear boletines

RF 24: Enviar boletines

Gestionar artículos

RF 25: Adicionar artículos

RF 26: Modificar artículos

RF 27: Mostrar artículos

RF 28: Eliminar artículos

2.4.2 Requisitos no funcionales.

Los requisitos no funcionales son las cualidades o propiedades que el producto debe tener. Estas propiedades constituyen las características que ayudan a que el producto sea atractivo, usable, rápido o confiable. Decretan cómo tiene que comportarse el sistema, qué cualidades debe tener o cuán rápido o grande debe ser. La propuesta de solución presenta los siguientes requisitos no funcionales:

RNF 1 Usabilidad

- En el *OpenCourseWare* podrán trabajar, desde usuarios con conocimientos avanzados de informática hasta usuarios inexpertos; lo cual será logrado a partir de una correcta organización de la información, con el empleo de menús, que proporcionan una navegación sencilla.
- El *OpenCourseWare* ha de ser interactivo y de fácil comprensión por los usuarios, de manera que aunque se tenga poca experiencia en informática sea de fácil uso, la finalidad de dicho portal es informar a los usuarios que lo visitan.

RNF 2 Hardware

- El sistema debe presentar las siguientes características:
 - *Hardware* mínimo para el servidor web:
 - 2 GB de memoria RAM.
 - 80 GB de disco duro.

RNF 2 Navegabilidad

- El menú de navegación estará disponible en todas las páginas.

RNF 3 Visualización en los navegadores

Deberá visualizarse de manera correcta en los navegadores más usados en la actualidad: Mozilla Firefox versión 23 o superior, Opera Chrome versión 3 o superior, Internet Explorer versión 10 o superior, Chrome versión 3 o superior y Safari en su versión 3 o superior sin perder el atractivo visual.

RNF 4 Legales

- Aplicación basada en una plataforma con licencia GNU/GPL.

RNF 5 Restricciones de diseño e implementación.

Utilizar los lenguajes de programación definidos durante la investigación: Como lenguaje para la implementación del sistema se empleará Javascript. También se hará uso del lenguaje SQL para realizar las consultas a la base de datos y como tecnología web. Se utilizará además, el lenguaje de programación PHP en su versión 5.3.8

- Utilizar el gestor de base de datos PostgreSQL en su versión 9.1.
- Utilizar el servidor web Apache en su versión 2.2.2.
- El intérprete de aplicaciones web: Navegador.

2.5 Modelo de Casos de Uso del sistema.

En el modelo de casos de uso, el sistema agrupa casos de uso y los actores que inicializan los mismos. Los actores del sistema son aquellos individuos u otros sistemas externos que interactúan con el *software*. El actor establece un rol en el cual varios individuos pueden jugar ese papel. Para el desarrollo de los componentes se identificaron los actores siguientes.

Actor	Descripción
Editor	Es el usuario que se dedica a controlar la gestión de los artículos y las listas de cursos.
Administrador	Es el usuario que además de contar con el acceso pleno y controlar todas las acciones del sistema, se encarga de la gestión de usuarios, de asignar los roles a los usuarios anónimos, la gestión de las notificaciones, la gestión de las categorías y de los servicios OAI-PMH.
Profesor	Es el usuario que se dedica a controlar todo el proceso editorial dentro del sistema, encargándose de la gestión de cursos y la gestión de los materiales.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

Usuario anónimo	Es la persona que solo puede autenticarse en el sistema para acceder a la información autorizada, además puede recuperar información.
Usuario autenticado	Es el usuario que podrá tener acceso a los cursos y artículos podrá contactar al administrador a través del correo electrónico.

Tabla 2: Actores del sistema.

2.5.1 Definición de los casos de uso.

Un caso de uso es una secuencia de interacción que se desplegará entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Posterior a la descripción de los actores del sistema se realiza el proceso de identificación de los casos de uso, con ayuda de los requisitos funcionales del sistema, los casos de uso identificados son:

- Autenticar usuario: permite a un usuario autenticarse en el sistema para que le sean otorgados los permisos necesarios para realizar las tareas específicas de acuerdo a su rol.
- Contactar administrador: permite contactar con el administrador a través de un correo electrónico.
- Gestionar cursos: permite agregar, editar y eliminar cursos.
- Gestionar materiales: permite crear, editar y eliminar materiales de un curso.
- Gestionar usuarios: permite crear, modificar, eliminar y listar los usuarios existentes en el sistema.
- Gestionar categorías: permite listar y editar las categorías; así como gestionar los campos y la presentación de estos.
- Gestionar listas de cursos: permite crear, editar, eliminar y mostrar cursos en una lista de cursos.
- Gestionar servicios OAI-PMH: permite generar un archivo XML con la descripción de un curso a partir de una petición HTTP.
- Gestionar boletines: permite crear y enviar boletines a usuarios.
- Gestionar artículos: permite crear, modificar, eliminar y listar los artículos.

2.5.2 Diagrama de casos de uso del sistema

El diagrama de casos de uso del sistema representa gráficamente los procesos y su interacción con los actores representando las funcionalidades del mismo.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

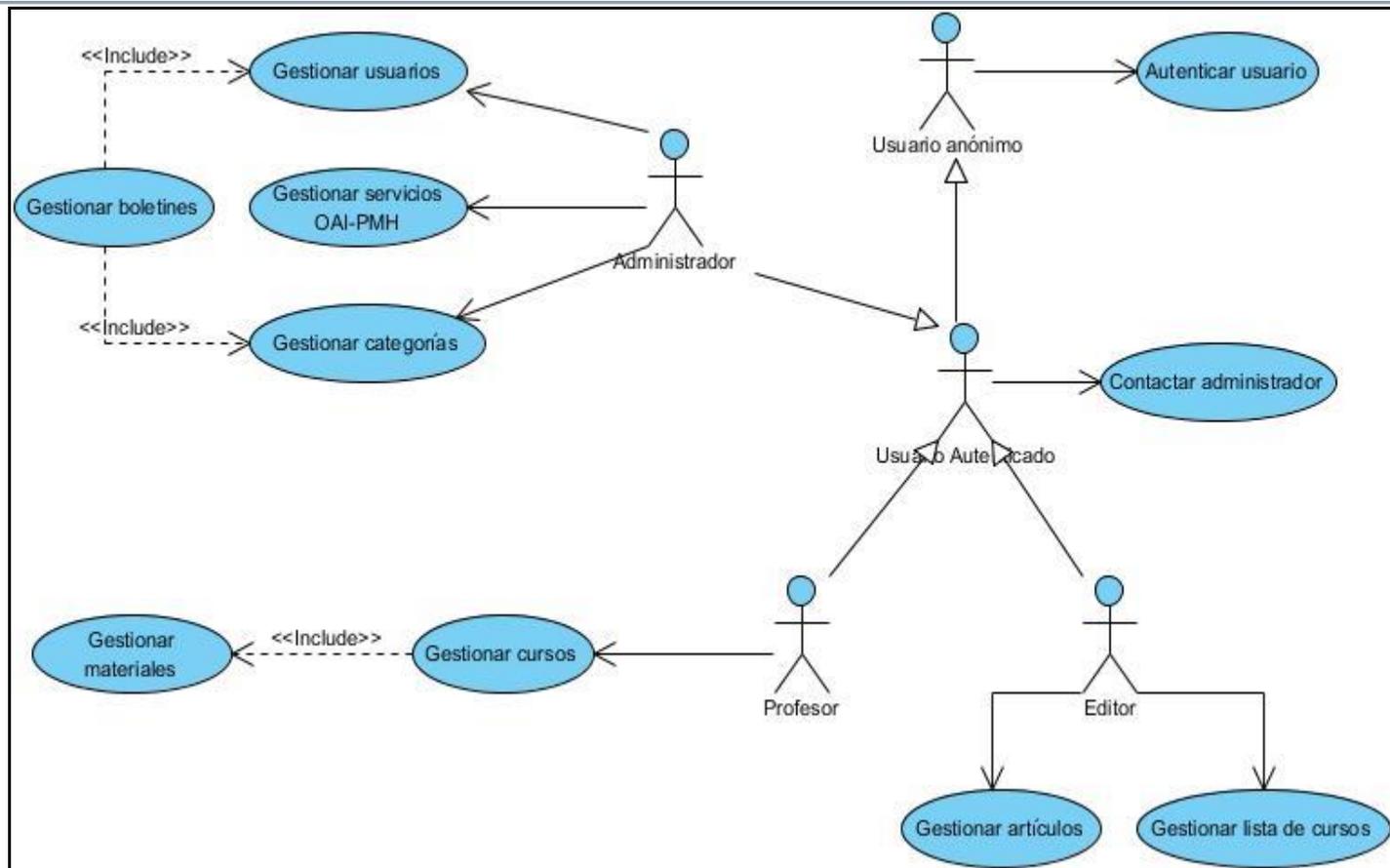


Figura 5: Diagrama de casos de uso del sistema.

2.5.3 Especificación de Casos de Uso.

A continuación se ofrece la especificación de algunos casos de uso que resultan críticos para la solución propuesta. Los restantes casos de uso se pueden encontrar en el Anexo 2.

Actores	Profesor: (Inicia) Agrega y edita los cursos en el sistema.
Resumen	El CU se inicia cuando el Profesor va a crear un curso.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	_____
Postcondiciones	El curso queda creado
Flujo de eventos	
Flujo básico	

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

	Actor	Sistema
1.	Se dirige al menú principal y da clic en la opción Gestionar Cursos	1.1 El sistema muestra las opciones Agregar cursos y Editar cursos.
2.	Selecciona una de las opciones	2.1 Si selecciona la opción "Agregar cursos" ir a la sección Agregar cursos. 2.2 Si selecciona la opción "Editar cursos" ir a la sección Agregar cursos.
Sección 1: Agregar cursos		
	Actor	Sistema
3	El profesor selecciona la opción "Agregar cursos"	3.1 El sistema muestra un formulario en el que solicita llenar los campos que posee el curso: Cursos, Categorías, Imagen, Profesor, Fundamentación, Conocimientos previos recomendados, Descripción general, Objetivos, Contenidos, Habilidades, Programa y Evaluaciones.
4	El profesor inserta los datos a llenar	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 El sistema verifica que el curso no haya sido agregado anteriormente. 4.3 El sistema notifica al profesor que creó un curso correctamente.
Flujos alternos		
	Actor	Sistema
4	a) El profesor procede a llenar los datos del formulario y da clic en la opción "Guardar".	b) El sistema muestra un mensaje "Curso X se ha creado".
	c) El profesor procede a llenar los datos del formulario y da clic en la opción "Vista previa".	d) El sistema muestra una vista previa del curso que se crea. e) El sistema va a la página donde se muestra el curso ya creado. Se termina el caso de uso.
Sección 2: Editar cursos		
	Actor	Sistema
5	El profesor selecciona la opción "Editar cursos".	5.1 El sistema muestra un listado con todos los cursos creados por el profesor.
6	El profesor selecciona el curso que desea Editar.	6.1 El sistema muestra un formulario con todos los datos del curso seleccionado para ser cambiados.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

		6.2 El sistema verifica si los datos cambiados son correctos.
Flujos Alternos		
	Actor	Sistema
6	a) El profesor procede a editar los datos del formulario y da clic en la opción "Guardar".	b) El sistema muestra un mensaje "Curso X ha sido actualizado".
	c) El profesor procede a llenar los datos del formulario y da clic en la opción "Vista previa".	d) El sistema muestra una vista previa del curso editado.
	e) El profesor no procede a llenar los datos del formulario y da clic en la opción "Eliminar". g) El profesor afirma	f) El sistema verifica que se desee realizar esta acción. h) El sistema muestra un mensaje "Curso X ha sido eliminado" g) El sistema va a la página donde se muestra el curso ya editado. Se termina el caso de uso.
Relaciones	CU Incluidos	Gestionar materiales. Ver CU Gestionar Cursos
Requisitos no funcionales		

Tabla 3: Especificación del caso de uso Gestionar Cursos.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

Crear Cursos

Cursos *

Categorías *

- Seleccione un valor - ▾

Imagen

No se ha seleccionado ningún archivo.

Los archivos deben ser menores que 2 MB.
Tipos de archivo permitidos: png gif jpg jpeg.

[Mostrar pesos de la fila](#)

Profesor

Fundamentación

Figura 6: Prototipo de interfaz de usuario del caso de uso Gestionar Cursos.

Objetivo	Gestionar el trabajo con las categorías	
Actores	Administrador: (Inicia) Lista, edita, gestiona los campos y la presentación de las categorías.	
Resumen	El CU se inicia cuando el Administrador	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	-----	
Postcondiciones	Quedan creadas las categorías	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El administrador se dirige al menú Administración y dentro de este busca en Estructuras el tipo de Taxonomía que tiene por nombre Categorías.	1.1 El sistema muestra un formulario con las opciones Lista, Editar, Gestionar campos, Gestionar presentación

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

2.	El administrador selecciona una de las opciones.	<p>2.1 Si selecciona la opción “Lista” ir a la sección “Lista”.</p> <p>2.2 Si selecciona la opción “Editar” ir a la sección “Editar”.</p> <p>2.3 Si selecciona la opción “Gestionar campos” ir a la sección “Gestionar campos”.</p> <p>2.4 Si selecciona la opción “Gestionar presentación” ir a la sección “Gestionar presentación”.</p>
Sección 1: Lista		
	Actor	Sistema
3	El administrador selecciona la opción “Lista”	El sistema muestra todas las categorías creadas por el administrador
Sección 2: Editar		
	Actor	Sistema
4	El administrador selecciona la opción “Editar”.	4.1 El sistema muestra un formulario con los campos pertenecientes a la categoría.
Flujos Alternos		
	Actor	Sistema
4	a) El administrador procede a modificar los datos del formulario y da clic en la opción “Guardar”.	b) El sistema muestra un mensaje “Los datos han sido guardados”.
	c) El administrador no procede a modificar los datos del formulario y da clic en la opción “Eliminar”.	<p>d) El sistema muestra un mensaje “La categoría ha sido eliminada”.</p> <p>e) El sistema va a la página donde están las opciones de menú. Se termina el caso de uso.</p>
Sección 3: Gestionar campos		
	Actor	Sistema
5	El administrador elige la opción “Gestionar campos”.	5.1 El sistema muestra un formulario en el que pide llenar los campos existentes en la categoría.
6	El administrador inserta los datos a llenar.	<p>6.1 El sistema verifica que todos los campos del formulario estén llenos.</p> <p>6.3 El sistema notifica al autor que los</p>

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

		campos fueran llenados satisfactoriamente.
Flujos Alternos		
	Actor	Sistema
5	a) El administrador procede a llenar los datos y da clic en la opción "Guardar".	b) El sistema muestra un mensaje "Los datos han sido guardados".
Sección 4: Gestionar presentación		
	Actor	Sistema
7	El administrador elige la opción "Gestionar presentación".	7.1 El sistema muestra un formulario con las opciones de presentación que se pide establecer.
8	El administrador inserta los datos a llenar	8.1 El sistema verifica que todos los campos del formulario estén llenos. 8.32 El sistema notifica al autor que los campos fueran llenados satisfactoriamente.
Flujos Alternos		
	Actor	Sistema
7	a) El administrador procede a llenar los datos y da clic en la opción "Guardar".	b) El sistema muestra un mensaje "Los datos han sido guardados".
Relaciones	CU Incluidos	Gestionar notificaciones <u>Ver CU Gestionar boletines.</u>

Tabla 4: Especificación del caso de uso Gestionar Categorías.

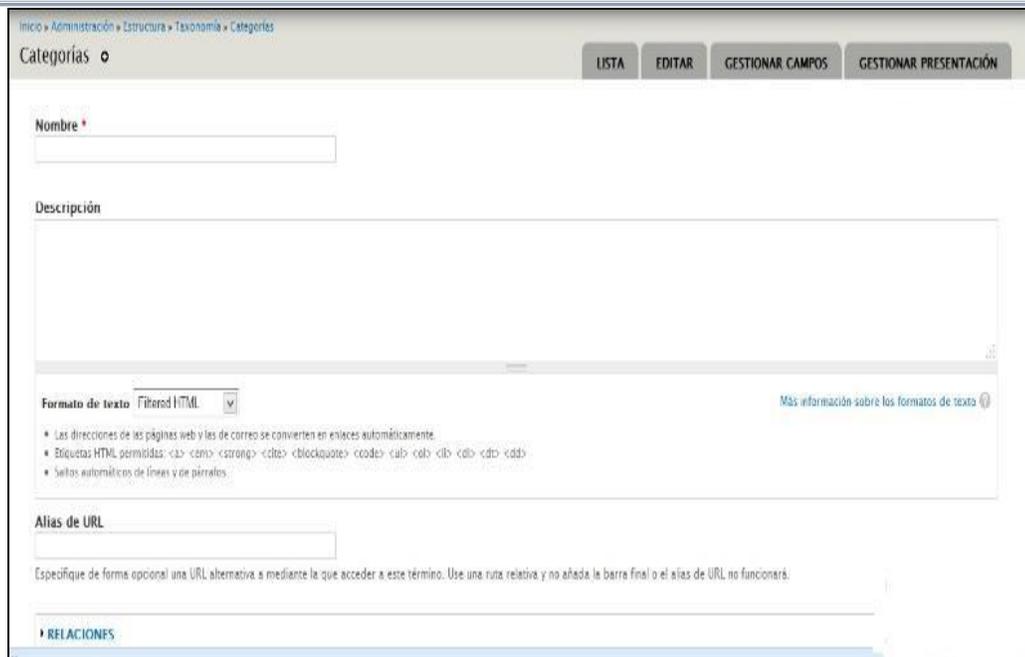


Figura 7: Prototipo de interfaz de usuario del caso de uso Gestionar Categorías.

2.6 Modelo de diseño.

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y para documentar el diseño del sistema de *software*. Es un producto de trabajo sistémico y compuesto que comprende todas las clases de diseño, subsistema, paquetes, colaboraciones y las relaciones entre ellos. En la fase de diseño se toman decisiones prácticas y precisas para dar alcance a los requerimientos funcionales y de calidad de un sistema. Se comprende mejor las relaciones existentes entre los requisitos no funcionales y las restricciones referidas a los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de interfaz de usuario entre otros. El modelo de diseño es utilizado como entrada en las actividades relacionadas a la implementación y representa a los casos de uso en el dominio de la solución.

2.6.1 Diagramas de clases de diseño.

Una clase del diseño es una abstracción de una clase o construcción análoga en la implementación del sistema. El lenguaje que se utiliza en dichas clases es el mismo que se emplea para la implementación del sistema; se especifican los atributos y las operaciones; se pueden realizar interfaces si tienen sentido para la programación y los métodos tienen correspondencia con los métodos de la programación.

En resumen, el modelo del diseño detalla la realización física de los casos de uso centralizándose en los requisitos funcionales, junto con otras restricciones relacionadas en el entorno de implementación, tienen un impacto en el sistema que se propone y sirve de abstracción a la implementación y al código fuente del sistema.

Estructura y funcionamiento de Drupal

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

Drupal cuenta con un grupo de clases o módulos que permiten a los usuarios finales interactuar con la aplicación, estos están organizados de acuerdo a sus funcionalidades. La distribución que constituye básicamente el CMS Drupal contiene 4 paquetes principales: *Themes*, *Includes*, *Modules* y *Scripts*, que contienen a su vez gran cantidad de archivos. En busca de una mayor comprensión de los diagramas de diseño web se modela primeramente el diagrama general de Drupal:

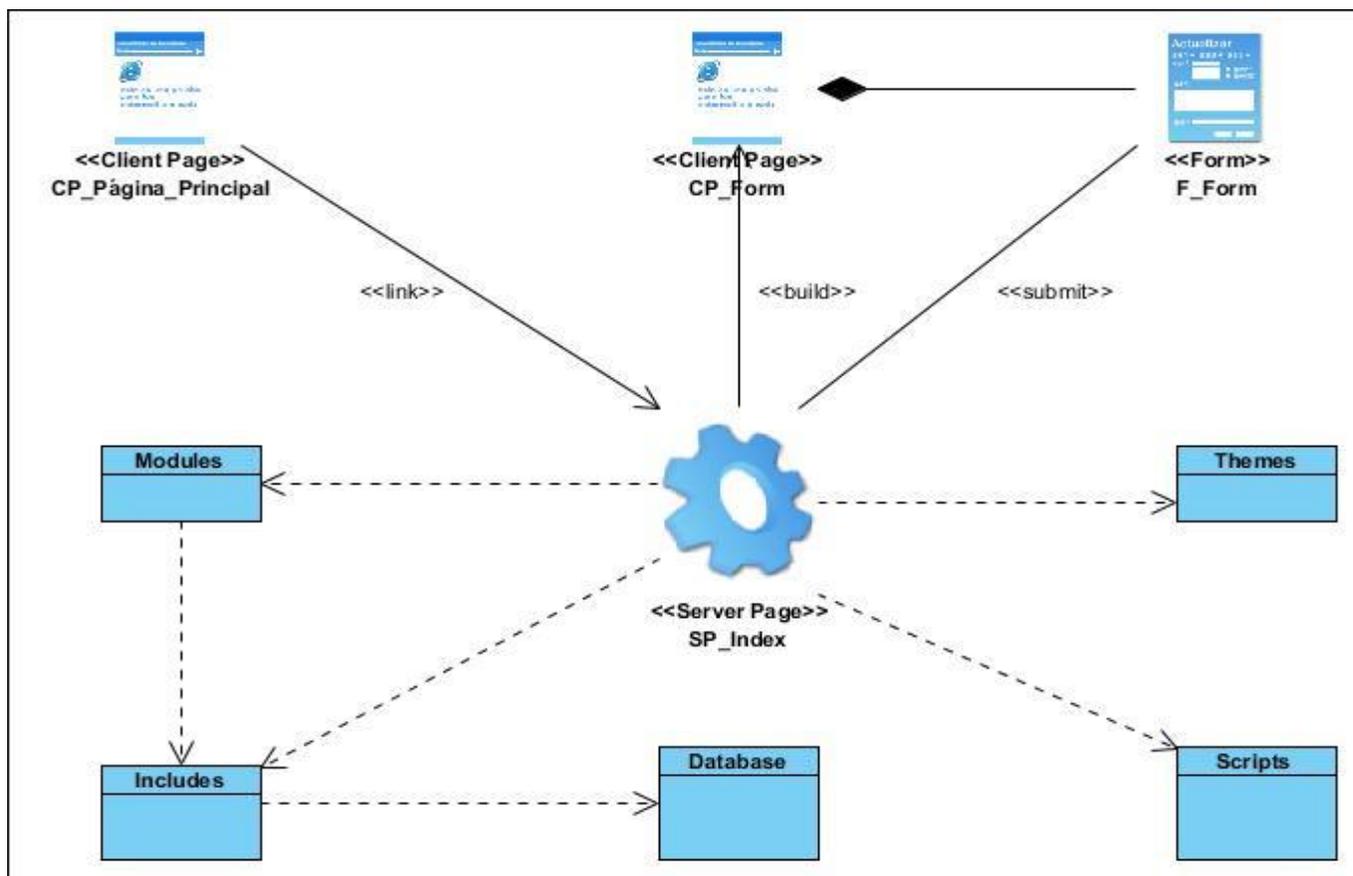


Figura 8: Diagrama de clases del diseño web de Drupal.

Includes: agrupa un conjunto de librerías en forma de archivos PHP con extensión *.inc*, que incluyen funciones comunes del sistema indispensables para su funcionamiento, como son las conexiones a la base de datos.

Modules: agrupa todos los módulos del núcleo que viabilizan las distintas funcionalidades del CMS, cada uno en su carpeta correspondiente. Nunca se deben modificar directamente los módulos, ni subir módulos adicionales a esta carpeta.

Scripts: agrupa utilidades adicionales que no utiliza Drupal directamente, pero que se pueden manejar desde la línea de comandos de *shell*. Por ejemplo, el script *password-hash.sh* permite obtener una contraseña codificada a partir de la contraseña original (en texto plano).

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

Themes: agrupa los temas que vienen con la distribución de Drupal. Los temas nuevos que se desean añadir, crear o modificar, se deben subir a `/sites/all/themes` o `/sites/default/themes`.

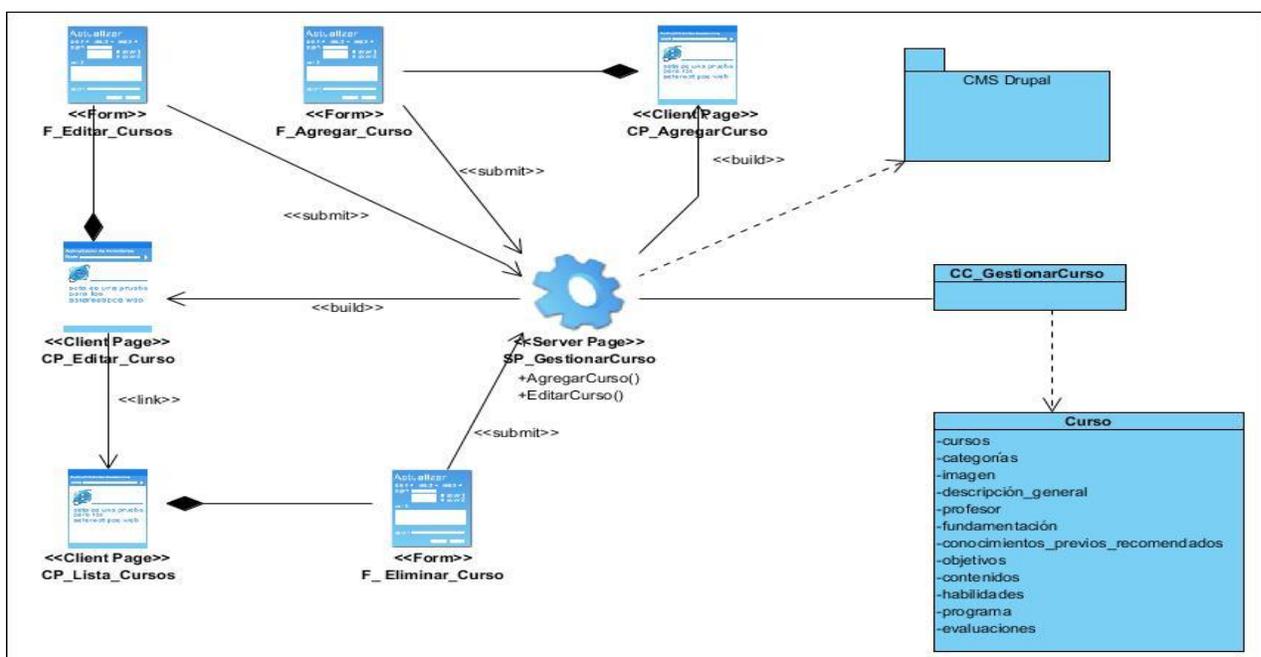
Index: El archivo `index.php` es la puerta de entrada al sistema. Cuando se carga una página del sitio web se está haciendo siempre una llamada a `index.php`.

Es importante señalar que el CMS Drupal contiene una única página servidora, que a través de funciones o ganchos (*hook*) se conecta con los módulos, los que a su vez se conectan a la base de datos a través de los *includes*, para finalmente mostrar la información requerida.

Luego de haber confeccionado el diagrama de clases del diseño por paquetes, se modela el diagrama de clases del diseño para cada subpaquete como se muestra a continuación. Los restantes diagramas se ubican en el Anexo 5.

Diagrama de clases de diseño: Gestionar cursos

Para agregar un curso la página servidora `SP_Gestionar_Curso` construye la página cliente `CP_Agregar curso`, la última se compone de un formulario `F_Agregar_Curso` donde se van a introducir los datos y que va a ser ejecutado por la página servidora. Para modificar un curso la página servidora construye la página cliente `CP_Editar_Curso` que obtiene la lista de cursos de la página cliente `CP_Lista_Cursos`, la primera se compone de un formulario `F_Editar_Curso` donde se van a modificar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora `CC_GestionarCurso`. Para eliminar un curso la página servidora construye la página cliente `CP_Lista Cursos`, la última se compone de un formulario `F_Eliminar_Curso` donde se van a eliminar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora.



CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

Figura 9: Diagrama de clases del diseño de Gestionar Curso.

Diagrama de clases de diseño: Gestionar Categorías

Para crear una nueva categoría la página servidora SP_GestionarCategoría construye la página cliente CP_GestionarCampos, la última se compone de un formulario F_GestionarCampos donde se van a introducir los datos y que va a ser ejecutado por la página servidora. Para editar una categoría la página servidora construye la página cliente CP_Editar, la última se compone de un formulario F_Editar donde se van a editar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora CC_Gestionar Categoría. Para listar las categorías la página servidora construye la página cliente CP_Lista, la última se compone de un formulario F_Lista donde se van a mostrar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora. Para gestionar la presentación de las categorías la página servidora construye la página cliente CP_GestionarPresentación, la última se compone de un formulario F_GestionarPresentación donde se van a mostrar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora.

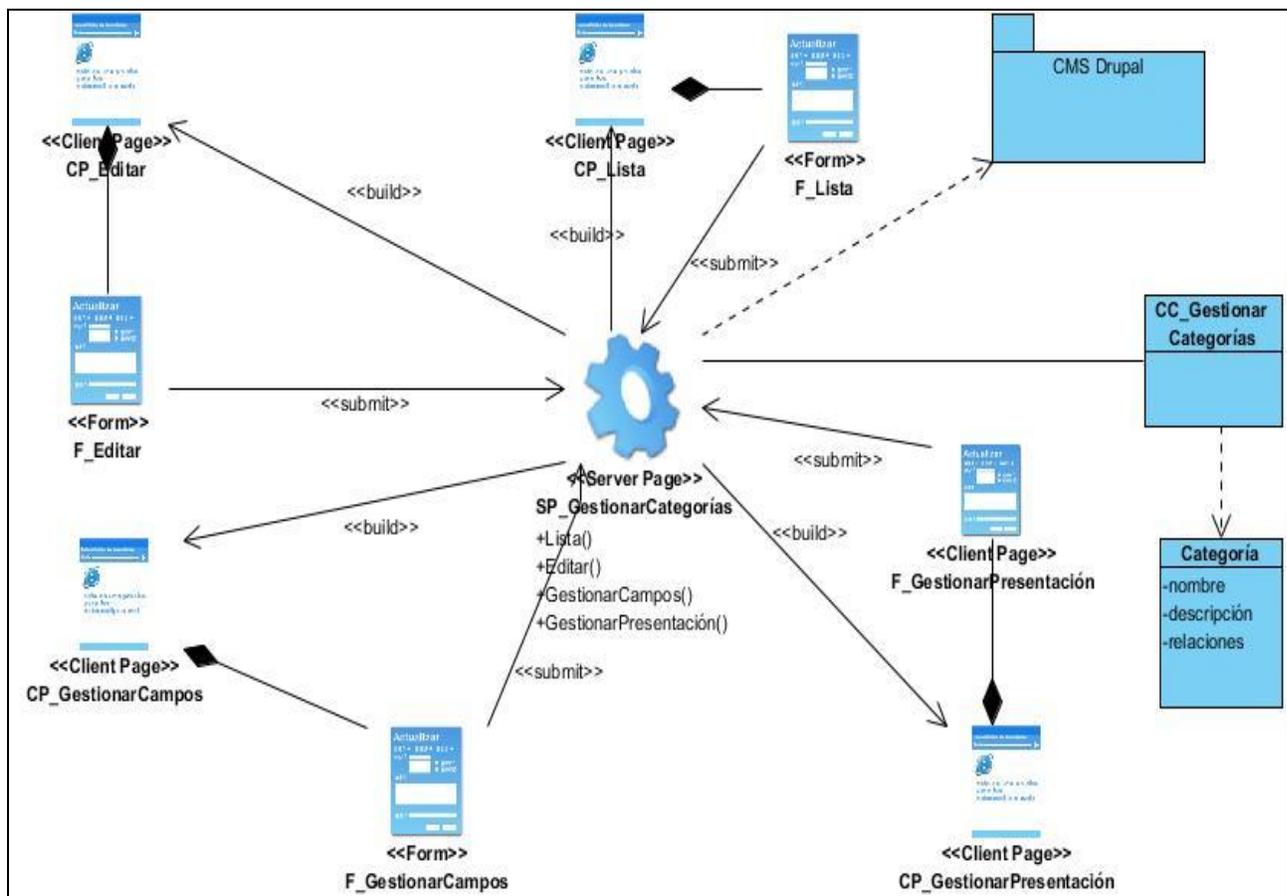


Figura 10: Diagrama de clases del diseño de Gestionar Categorías.

2.7 Diagramas de interacción.

Modelan el comportamiento dinámico del sistema; el flujo de control en una operación. Representa la interacción entre objetos; los cuales interactúan a través de mensajes para cumplir ciertas tareas. Estas interacciones proporcionan un comportamiento y en varias ocasiones implementan un Caso de Uso. Los diagramas de interacción se pueden clasificar en diagramas de secuencia y en diagramas de colaboración.

Es decir que estos diagramas describen como un grupo de objetos colaboran entre sí para lograr un fin mediante el paso de mensajes entre ellos dentro del caso de uso.

2.7.1 Diagrama de secuencias.

Los diagramas de secuencias revelan la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela por cada sección de la descripción de los casos de uso, el diagrama de secuencia contiene los detalles de implementación del escenario, incluyendo los objetos y clases que se aplican para implementar el escenario además de los mensajes entre los objetos.

El diagrama de secuencia está constituido por la línea de vida que representa la existencia de un objeto a lo largo de un período de tiempo. El foco de control que es el rectángulo delgado que representa el período de tiempo durante el cual el objeto ejecuta una acción. Para la realización de los casos de usos en el diseño se utilizan los diagramas de secuencia debido a que representan el flujo de acciones con mayor claridad.

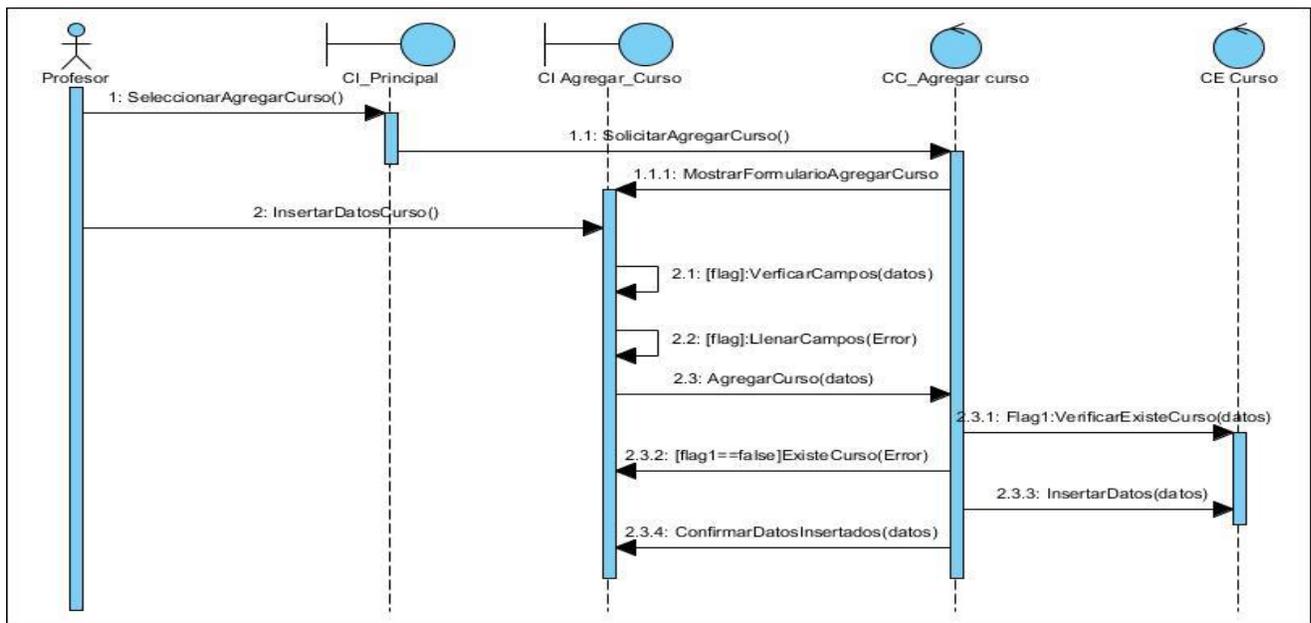


Figura 11: Diagrama de secuencia RF Agregar curso.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

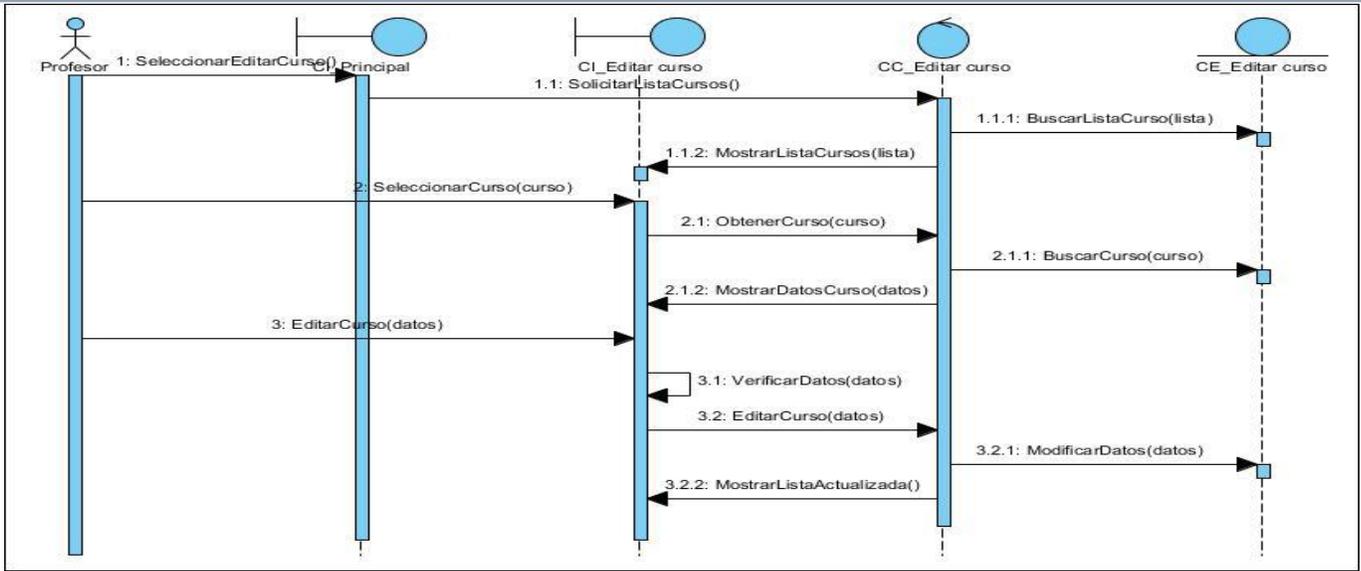


Figura 12: Diagrama de secuencia RF Editar curso.

2.8 Arquitectura del sistema.

Para la implementación del sitio web *OpenCourseWare* con el Sistema de Gestión de Contenidos Drupal se utiliza una arquitectura en capas como se muestra en la Figura 13:

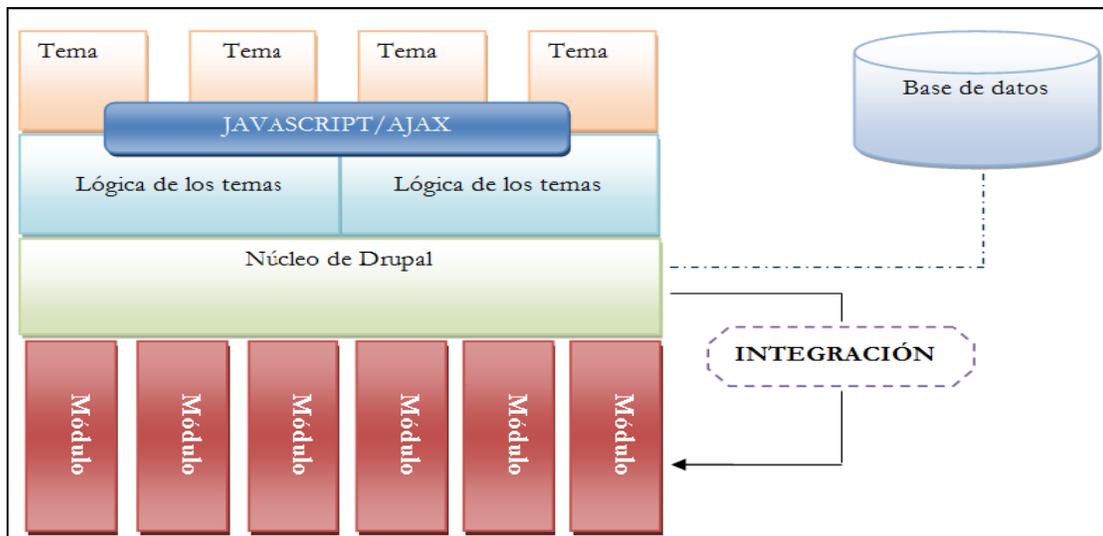


Figura 13: Arquitectura de Drupal (DRUPAL, 2011).

Cada módulo está compuesto por funciones, entre ellas aquellas llamadas *hooks*. Los *hooks* se encargan de notificar al sistema qué hacer y cuándo, de esta forma Drupal ejecuta los módulos solo cuando se necesita. Las estructuras que se manipulan en las funciones son las entidades (*entity*, el principal concepto de Drupal 7). El nodo deja de ser lo fundamental y se convierte en una entidad, al igual que el usuario, la taxonomía y el comentario. Las entidades tienen *bundles* y un *bundle* contiene campos (*fields*). La presentación es manejada por el sistema de *themes*.

2.9 Descripción de patrones utilizados.

2.9.1 Patrones arquitectónicos utilizados.

Los patrones arquitectónicos representan el nivel más alto dentro del sistema de patrones y expresan el esquema de la estructura fundamental de la organización para sistemas de *software*. Proveen un conjunto de subsistemas predefinidos, especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Cada patrón ayuda a lograr una propiedad específica del sistema global como es la adaptabilidad de la interfaz de usuario (ALMEIRA, 2011).

Para la implementación del sitio *OpenCourseWare*, se utilizará el CMS Drupal y por tanto el patrón de diseño usado sería el Modelo Vista Controlador (MVC), que trae Drupal incluido. Este es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes diferentes. El MVC es observado con frecuencia en las aplicaciones web, donde el modelo no es más que el Sistema Gestor de Base de Datos y la lógica del negocio, la vista es la página HTML y el código que provee de datos dinámicos a la página y el controlador es aquel responsable de recibir los eventos de entrada desde la vista.

Modelo: representa la parte de la aplicación que implementa la lógica del dominio de datos de la aplicación, en la propuesta de solución se evidencia cuando se realizan consultas a la base de datos que hacen uso de la capa de abstracción que tiene Drupal.

Vista: representa la interfaz de usuario, es utilizada por la capa controladora para la interacción con el usuario, en la propuesta de solución. La vista se evidencia en el sistema en “*Themes*”, “*Themes Engines*”, “*Javascript/Ajax*”.

Controlador: se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista, en la propuesta de solución el controlador es “*Drupal Core*” y “*Modules*” que representa a los módulos donde “*Hook*” son funciones que facilitan la comunicación entre el “*Drupal Core*” y “*Modules*”.

2.9.2 Patrones de diseño utilizados.

Un patrón de diseño es un esquema para ofrecer una solución a un determinado tipo de problema con el que nos encontramos con cierta frecuencia en el desarrollo de *software* (ARA, 2013).

Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas *software*.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Drupal utiliza patrones de diseños conformemente orientados a objetos como los patrones *Gang of Four* (GoF) que posibilitan la construcción de un diseño elegante y robusto. Estos patrones son clasificados según el propósito para el cual han sido especificados (MARTÍN, 2009). En el desarrollo del sitio web se utilizaron los siguientes patrones de diseño que son utilizados por Drupal (GAMMA, 2005):

Patrones estructurales: solucionan problemas de composición (agregación) de clases y objetos.

- **Bridge** (Puente): La capa de abstracción de bases de datos de Drupal se aplica de una forma similar al patrón de diseño bridge. En el desarrollo de la solución propuesta los módulos fueron escritos de forma independiente al sistema de base de datos utilizado. La información almacenada en la base de datos puede modificarse sin la necesidad de modificar el código de algún módulo.
- **Decorator** (Decorador): El CMS Drupal hace amplio uso del patrón decorador. Mediante el empleo de diferentes *hook* permite, que módulos arbitrarios extiendan el comportamiento de otros nodos. Esta característica posibilita añadir comportamientos a los nodos sin la necesidad de implementar una nueva subclase.

Patrones de comportamiento: solucionan problemas respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

- **Command** (Comando): encapsula una petición en un objeto, permitiendo así parametrizar a los clientes con diferentes peticiones, hacer cola o llevar un registro de las peticiones y poder deshacer las peticiones. En el desarrollo de la solución propuesta fue empleado el patrón *command* a fin de que los módulos implementados no tengan que definir cada *hook*, sino sólo los que vayan a aplicar.
- **Chain of Responsibility** (Cadena de responsabilidad): este patrón lo sigue el sistema de menús de Drupal. En la propuesta de solución, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados y qué función se llama para hacer el trabajo, siguiendo así el flujo de trabajo de este patrón hasta que un módulo se encarga de la petición, un módulo niega el acceso para el usuario o la cadena se ha agotado.
- **Observer** (Observador): es el encargado de definir una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Al igual que en el caso del patrón de diseño *Decorator* este módulo es utilizado para llevar a cabo la extensión de los componentes internos de Drupal a través de los correspondientes *hooks*.
- **Patrón de reflexión** (*Reflection Pattern*): es el encargado de describir el comportamiento de todo sitio basado en Drupal. Gracias a este patrón el sistema se convierte en una aplicación muy

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

adaptable a diferentes entornos puesto que permite prácticamente la modificación de todo su comportamiento a través de los módulos instalables, sin la necesidad de modificar el núcleo.

2.10 Diseño de la base de datos.

Generar las tablas que modelan los registros en los que se guarda la información del sistema es el principal objetivo del diseño de la base de datos. Para recuperar de forma rápida y eficiente los datos, es necesario que la información que se almacene no sea redundante. A continuación se muestra el diagrama de clases persistentes y el modelo de datos que son los que darán solución a la parte del diseño.

2.10.1 Modelo de datos.

En este modelo solo se representan las nuevas entidades que fueron creadas para dar solución al problema que se modela.

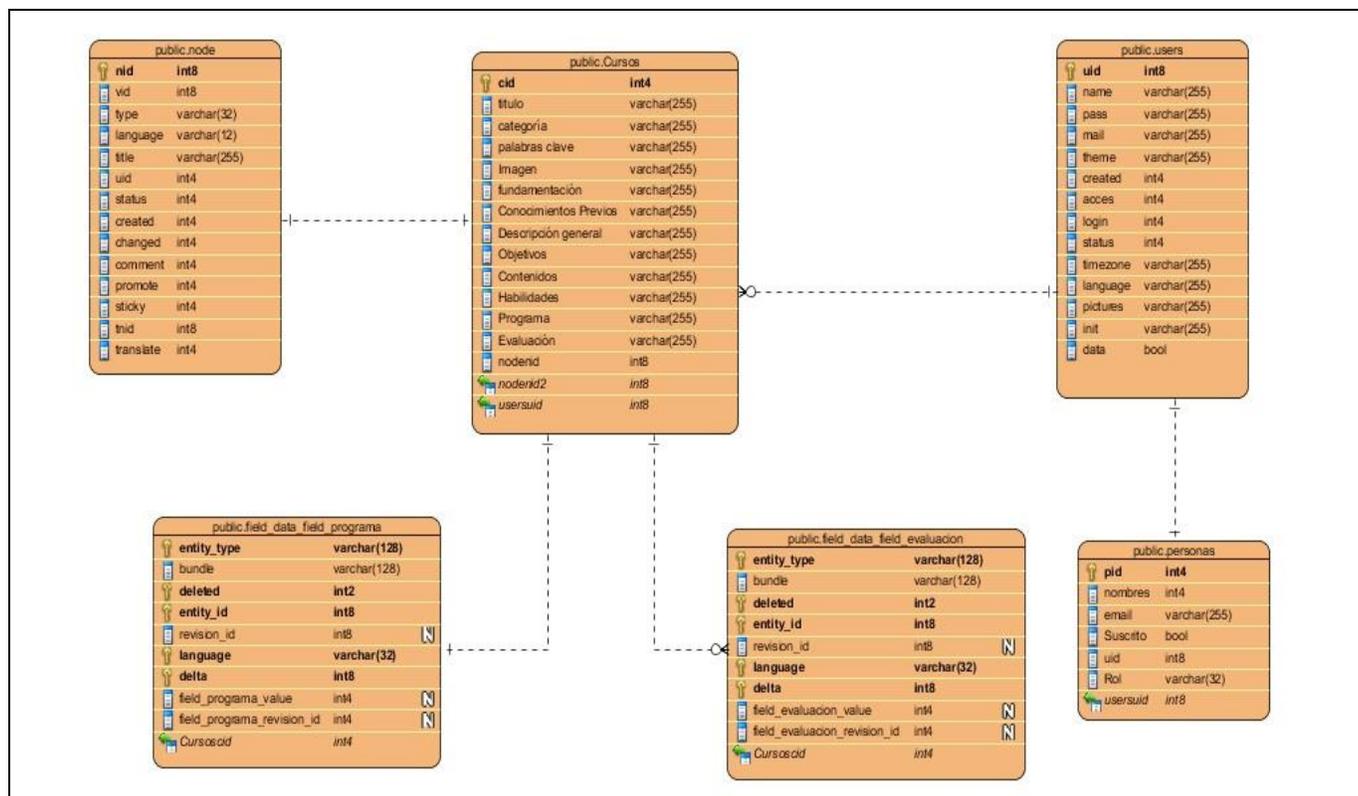


Figura 14: Modelo de datos del sistema.

A continuación se muestra una tabla con la descripción de una de las tablas del tipo de contenido Cursos que se empleó para el desarrollo de la aplicación:

Nombre: <i>public.Cursos</i>		
Descripción: Almacena los datos de los cursos		
Atributo	Tipo	Descripción

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

título	varchar	Nombre del curso
categorías	varchar	Categoría a la que pertenece el curso
palabras claves	varchar	Palabras claves que comprende el curso.
imagen	varchar	Imagen que se le adjunta al curso
fundamentación	varchar	Fundamentación del curso a crear
Conocimientos previos	varchar	Conocimientos previos que se deben poseer para optar por el curso.
Descripción general	varchar	Descripción general del curso
Objetivos	varchar	Objetivos que persigue el curso
Contenidos	varchar	Contenidos que tiene el curso
Habilidades	varchar	Habilidades que pretende desarrollar el curso
Programa	varchar	Programa que comprende el curso
Evaluación	varchar	Evaluación que se obtiene del curso

Tabla 5: Descripción de la tabla *public.Cursos*.

2.11 Modelo de despliegue.

El diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de *hardware* (nodos) y muestra cómo los elementos y artefactos del *software* se trazan en esos nodos. Este diagrama permite indicar la situación física de los componentes lógicos desarrollados. Cada *hardware* es representado como un nodo; un nodo es un elemento donde se ejecutan los componentes, entre ellos existen relaciones que son representados como medios de comunicación tales como HTTP, TCP/IP, etc.

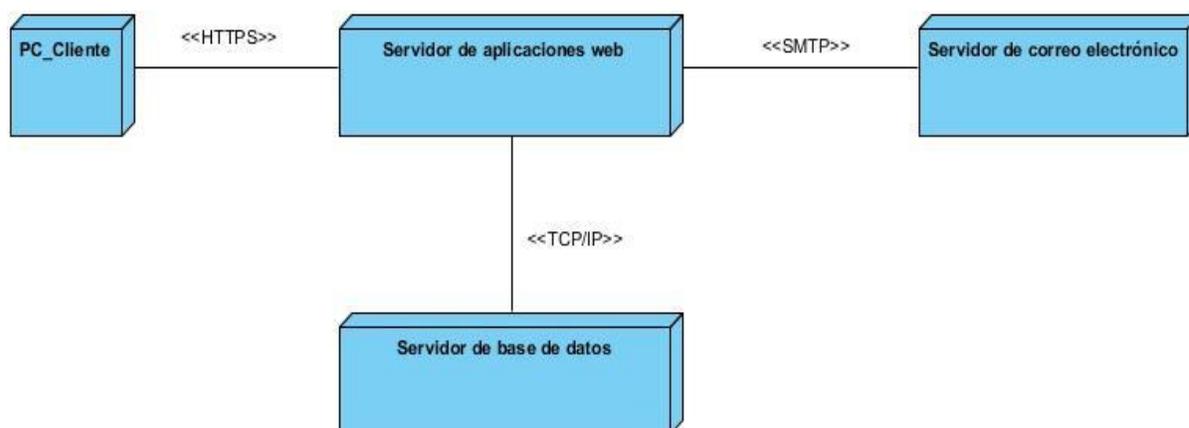


Figura 15: Diagrama de despliegue.

PC Cliente: simboliza el ordenador personal del cliente que se conecta al Servidor de aplicaciones web. El mismo se comunica al servidor mediante el Protocolo de Transferencia de Hipertexto.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA ACERCA DEL DOMINIO CONCEPTUAL Y TECNOLÓGICO

Servidor de aplicaciones web: simboliza el servidor sobre el cual corre la aplicación web. Este accede al Servidor de Base de Datos para el manejo de información.

Servidor de base de datos: es la representación de la base de datos perteneciente a la aplicación.

Servidor de correo electrónico: simboliza al servidor a través del cual se realiza la gestión de los boletines.

Capítulo 3: Implementación y prueba del sistema

3.1 Introducción

El presente capítulo comprende la implementación de los componentes del sitio web *OpenCourseWare*, para ello se describe el modelo de componentes. Además, se describen los casos de prueba a los que fueron sometidas las funcionalidades del sistema en cada una de las iteraciones. También se detallan las pruebas realizadas a los casos de pruebas y a la aplicación en funcionamiento.

3.2 Implementación.

El modelo de implementación permite organizar los componentes de acuerdo con los mecanismos de estructuración disponible en el entorno de implementación y en el lenguaje utilizado, y muestra la dependencia entre componentes.

3.2.1 Diagrama de componentes.

El diagrama de componentes modela las dependencias lógicas entre componentes de *software*. Estos diagramas son utilizados para detallar la vista de implementación estática de un sistema determinado. Los mismos cuentan con un nivel mucho más profundo de abstracción en correspondencia con el diagrama de clases y representan una o varias clases, interacciones o colaboraciones (ULTIMATE, 2013).

Los componentes físicos incluyen archivos, bibliotecas, ejecutables o paquetes. Estos diagramas permiten modelar y documentar la arquitectura haciendo posible la visualización de la organización y las dependencias entre un conjunto de componentes. Los componentes, por su parte, son fragmentos modulares del sistema que pueden desplegarse y reemplazarse. También encapsulan implementación y un conjunto de interfaces proporcionado la realización de los mismos.

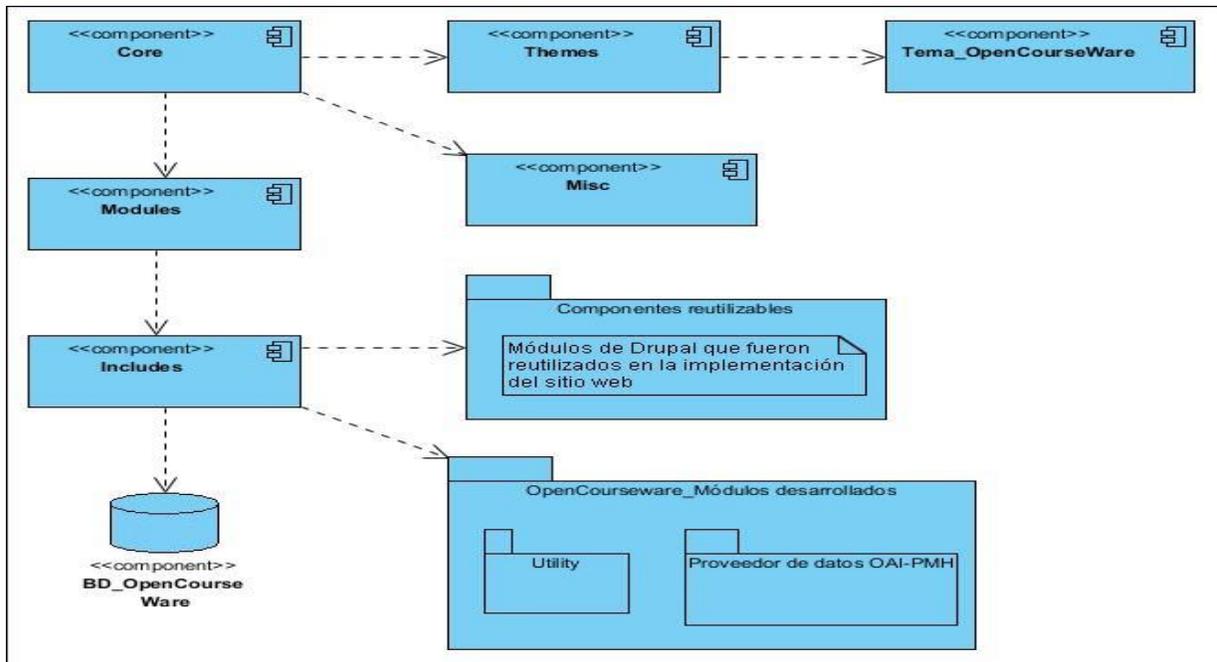


Figura 16: Diagrama de componentes.

La tabla que se muestra a continuación describe cada uno de los componentes representados en el diagrama:

Componentes	Descripción
<i>Themes</i>	En este componente se encuentra toda la capa de presentación y diseño.
<i>Tema_OpenCourseWare</i>	Nuevo tema creado para el sitio web <i>OpenCourseWare</i> .
<i>Core</i>	Constituye el punto de inicio del sitio, a partir de esta entrada se invocan los diferentes módulos del CMS.
<i>Modules</i>	Contienen los módulos del CMS Drupal
<i>Include</i>	Este componente contiene un conjunto de librerías en forma de archivos PHP que contiene funcionalidades imprescindibles para el funcionamiento de Drupal.
<i>Misc</i>	Incluye elementos relacionados con el diseño y funcionamiento (imágenes, archivos Javascript, etc.)
<i>OpenCourseWare_Módulos Desarrollados</i>	Contiene los módulos desarrollados para el <i>OpenCourseWare</i> .
<i>BD_OpenCourseWare</i>	Es la base de datos del sitio web.

Tabla 6: Descripción de componentes.

3.3 Estándares de codificación.

Los estándares de codificación son por lo general normas que se deben seguir para realizar la elaboración del código fuente. La principal ventaja que adjudica la utilización de estos estándares es que permite a otros programadores entender el código generado. Los mismos delimitan buenas prácticas de programación para lograr un código robusto, lo que ayuda en la calidad del *software*.

En el desarrollo de la presente solución se decide seguir los estándares planteados por la comunidad de Drupal que facilitan la legibilidad y estructuración del desarrollo de código para este lenguaje. A continuación se relacionan los mismos:

- Se utilizan como principios para el manejo de las etiquetas de apertura y cierre de código PHP las siguientes:
 - Utilización de `<?php ?>` para delimitar el código PHP.
 - La etiqueta de apertura simplificada, `<? ?>` nunca debe usarse.
- Para la declaración de funciones se tiene que estas deben tener el nombre del grupo o módulo como prefijo, para evitar conflictos de nombres entre funciones de distintos módulos. Al escribir una función que utilice valores por defecto para algunos parámetros, deben listarse estos parámetros.
- Se coloca un espacio entre el (tipo) y la variable en una transformación. Por ejemplo, `(int) $mynumber`.
- Cuando se llama a un método constructor sin argumentos, deben incluirse siempre en la llamada los paréntesis del método.
- Se emplea siempre un espacio entre el punto y las partes concatenadas de una cadena para mejorar la legibilidad. Al concatenar variables simples se pueden usar comillas dobles y agregar la variable dentro. En otro caso, es necesario usar comillas simples. Al usar el operador de “concatenación-asignación” (`‘.=’`), también es necesario incluir un espacio antes y después del operador.
- Las clases son nombradas usando la nomenclatura “*CamelCase*”. Los métodos y propiedades de clases deben usar “*lowerCamelCase*”.
- Los nombres de las funciones estarán en minúsculas y basados en el nombre del módulo del que forman parte. Los guiones bajos se utilizarán para separar las palabras descriptivas del nombre. Después del nombre del módulo, la función debe nombrarse con el verbo y el objeto a los que hace referencia.
- El nombre de un módulo no debe contener guiones bajos para evitar conflictos de espacios de nombres (ANDALUCÍA, 2013).

3.4 Pantallas de la aplicación.



Figura 17: Pantalla principal del sistema.



Figura 18: Vista de un curso creado.

3.5 Validación del sistema.

3.5.1 Pruebas.

Las pruebas de *software* son aquellos procedimientos que se efectúan para contrastar la calidad de un producto de *software* y pueden ser aplicadas periódicamente. Las mismas tienen como objetivo fundamental la identificación de posibles errores que se presenten en la aplicación. Existen varias estrategias de pruebas que suelen ser utilizadas por los equipos de probadores, dentro de las que se pueden mencionar:

Pruebas funcionales

Se denominan pruebas funcionales o *Functional Testing*, a las pruebas de *software* que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre esta el paso siguiente es el pase a producción (B., 2009).

Para la realización de estas pruebas se utilizó la técnica de caja negra. Este tipo de prueba se centra en los requisitos funcionales del *software* y permite obtener entradas que prueben todos los requisitos funcionales del programa. Con este tipo de pruebas se intenta encontrar errores de interfaz, en estructuras de datos o en accesos a bases de datos externas, de rendimiento, entre otros que puedan atentar contra la integridad del sistema.

Para el cumplimiento de este tipo de prueba es necesario diseñar un conjunto de casos de prueba que se utilizan para verificar el buen funcionamiento de las diferentes funcionalidades del sistema y detectar las no conformidades que puedan existir.

A continuación se describen algunos de los casos de prueba para algunos requisitos con prioridad Alta.

Escenario	Respuesta del sistema	Flujo central
EC 1.1 Agregar curso correctamente.	El sistema guarda los datos correctamente y muestra un mensaje de confirmación "Curso X se ha creado".	<ol style="list-style-type: none"> 1. El usuario con permisos de profesor después de autenticado en el sistema accede al menú principal y se dirige a la opción del menú Gestionar cursos y selecciona Agregar cursos. 2. El sistema muestra la interfaz para crear el nuevo curso. 3. El usuario con permisos de profesor introduce los datos y da clic en el botón Guardar. 4. El sistema verifica que los datos estén correctos y los campos obligatorios completos.
EC 1.1.1 Agregar curso incorrectamente	El sistema no guarda los datos y muestra los mensajes de errores correspondientes: "El curso X existe ya en el sistema." Para los campos obligatorios que	<ol style="list-style-type: none"> 5. El sistema muestra el formulario para ingresar nuevamente los datos del nuevo curso, y los mensajes de error correspondientes a los campos obligatorios vacíos o incorrectos. 6. El usuario con permisos de profesor ingresa los datos correctos y da clic al botón Guardar. 7. El sistema verifica que los datos estén correctos y los campos obligatorios completos. 8. El sistema almacena los datos del nuevo curso.

	estén vacíos se muestra el mensaje "El campo X es obligatorio".	
EC 2.1 Mostrar datos de usuario.	El sistema muestra los usuarios según los criterios de búsqueda seleccionados	<ol style="list-style-type: none"> 1. El usuario con permisos de administrador después de autenticado en el sistema accede al menú de administración y se dirige a la opción del menú Personas. 2. El sistema muestra una nueva interfaz. 3. El usuario con permisos de profesor selecciona las opciones para mostrar los cursos y da clic en el botón Filtro. 4. El sistema realiza la búsqueda y muestra los usuarios en la parte inferior de la interfaz.
EC 2.2 Editar curso correctamente.	El sistema guarda los datos correctamente y muestra un mensaje de confirmación: "Se han guardado los cambios."	<ol style="list-style-type: none"> 1. El usuario con permisos de Profesor después de autenticado en el sistema accede al menú principal, se dirige a la opción del menú Editar cursos, selecciona del listado de cursos que muestra el sistema el curso que desea editar. 2. El sistema muestra la interfaz para editar el curso. 3. El usuario con permisos de profesor introduce los datos y da clic en el botón Guardar.
EC 2.2.1 Editar curso incorrectamente.	El sistema no guarda los datos y muestra los mensajes de errores correspondientes: Para los campos obligatorios que estén vacíos se muestra el mensaje "El campo X es obligatorio".	<ol style="list-style-type: none"> 4. El sistema muestra nuevamente el formulario para editar correctamente los datos del curso, y los mensajes de error correspondientes a los campos obligatorios vacíos. 5. El usuario con permisos de profesor edita los datos correctamente y da clic al botón Guardar.

Tabla 7: Caso de prueba Gestionar curso.

Escenario	Respuesta del sistema	Flujo central
	El sistema guarda los datos correctamente y muestra un mensaje de confirmación.	<ol style="list-style-type: none"> 1. El usuario con permisos de administrador después de autenticado en el sistema accede al menú de administración y busca allí el tipo de taxonomía que tiene por nombre Categorías. 2. El sistema muestra un formulario con diferentes opciones. 3. El usuario con permisos de administrador escoge la opción Gestionar campos e introduce los datos en los campos correspondientes. 4. El sistema verifica que los datos estén correctos y los campos obligatorios llenos.
EC 2.1.1 Gestionar campos incorrectamente		<ol style="list-style-type: none"> 5. El sistema muestra el formulario para ingresar nuevamente los datos de la nueva categoría, y los mensajes de error correspondientes a los campos obligatorios vacíos. 6. El usuario con permisos de administrador ingresa

		<p>los datos correctos y da clic al botón Guardar.</p> <p>7. El sistema verifica que los campos obligatorios completos.</p> <p>8. El sistema almacena los datos de la nueva categoría.</p>
	<p>El sistema no guarda los datos y muestra el mensaje de error correspondiente: "El campo Nombre es obligatorio".</p>	
	<p>El sistema muestra el listado de categorías existentes.</p>	<p>1. El usuario con permisos de administrador después de autenticado en el sistema accede al menú de administración y se dirige al tipo de taxonomía Categorías.</p> <p>2. El sistema muestra un formulario con varias opciones.</p> <p>3. El usuario con permisos de administrador selecciona la opción Lista.</p> <p>4. El sistema muestra las categorías existentes en el sistema.</p>

Tabla 8: Caso de prueba Gestionar categoría.

En el gráfico que se muestra a continuación (ver, Figura 19), se brinda información sobre el total de no conformidades encontradas y las que se resolvieron por cada iteración. Para un total de 28 requisitos funcionales se detectaron 9 no conformidades en la primera iteración las cuales fueron resueltas satisfactoriamente, y en la segunda iteración se redujeron las no conformidades solo a 3 que también fueron resueltas, número que se redujo hasta la tercera iteración donde no se obtuvieron no conformidades.

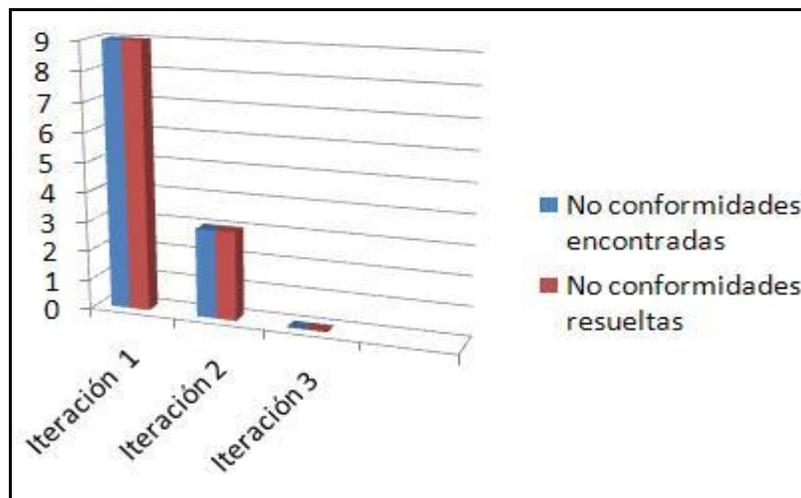


Figura 19: Cantidad de no conformidades por iteración.

Pruebas de Seguridad

La prueba de seguridad intenta verificar que los mecanismos de protección incorporados en el sistema lo protegerán, de hecho, de accesos impropios. (PRESSMAN, 2010)

Para la realización de pruebas de seguridad se escogió la herramienta *Acunetix* la cual es una herramienta de auditoría web que permite identificar diversos tipos de vulnerabilidades. La misma presenta una interfaz intuitiva y sencilla con la que permite comprobar un sitio web en busca de vulnerabilidades como *SQL Injection* y *Cross Site Scripting (XSS)* en formularios, páginas de *login*, contenido dinámico, etc.

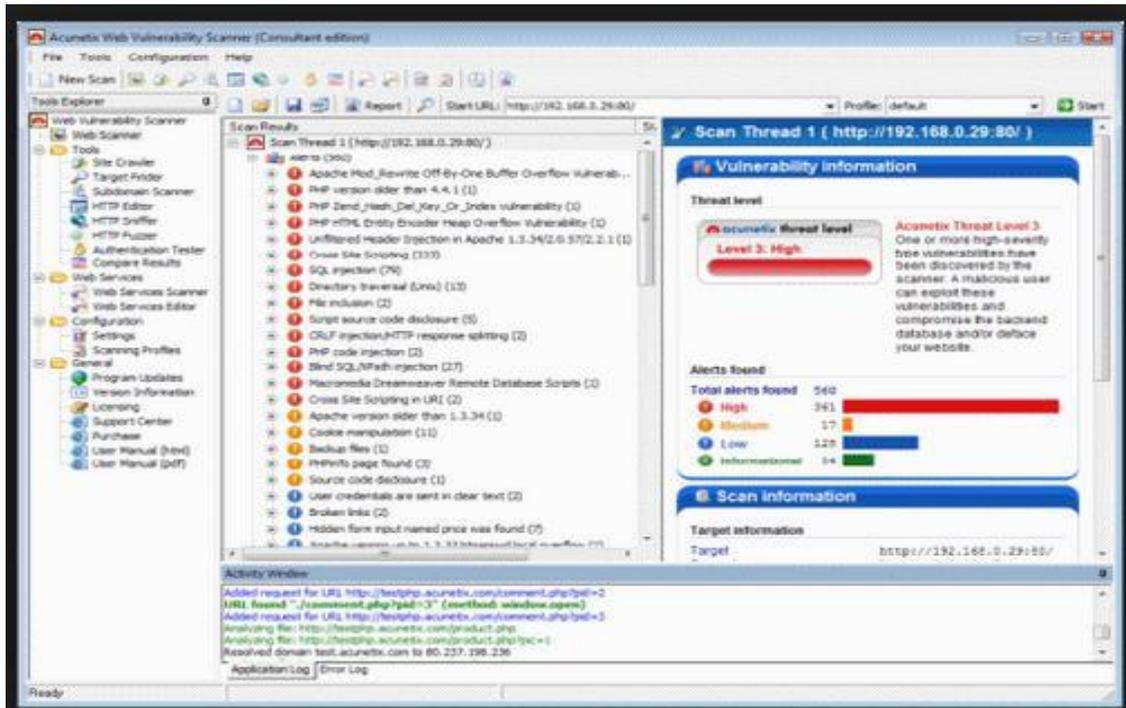


Figura 20: Interfaz gráfica de la herramienta de seguridad Acunetix.

Se realizaron dos iteraciones de pruebas con la herramienta las cuales arrojaron los resultados expuestos en las Figuras 21 y 22.



Figura 21: Resultados de la primera iteración de pruebas de seguridad.

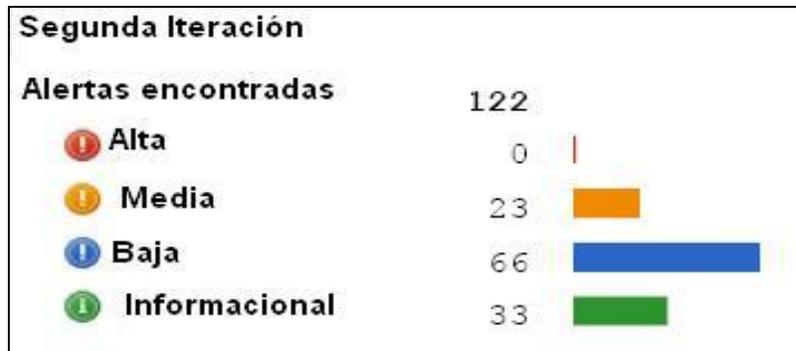


Figura 22: Resultados de la segunda iteración de pruebas de seguridad.

De forma general, en la primera iteración se detectaron 403 alertas, de ellas 310 de prioridad baja, 39 de prioridad media y 54 informativas; mientras que en la segunda iteración se redujeron a 122 las alertas encontradas de estas 23 de prioridad media, 66 de prioridad baja y 33 informativas. Entre las no conformidades de prioridad baja se encontraron: ficheros del servidor Apache vulnerables a ataques de fuerza bruta y *cookies* de sesión sin el indicador de seguridad. Las no conformidades de prioridad media están relacionadas con el auto completamiento de los formularios, se corrigió esta no conformidad. El resto deja al descubierto información acerca de una página de error que contiene la versión del servidor web y una lista de los módulos habilitados en este servidor.

Pruebas de rendimiento

El servicio de pruebas de rendimiento de *software* se centra en determinar la velocidad con la que el sistema bajo pruebas realiza una tarea en las condiciones particulares del escenario de pruebas. Este servicio ayuda a su organización a detectar los cuellos de botella de su aplicación, antes de que, sus usuarios sufran un mal rendimiento, con la consecuente pérdida económica y frustración de sus clientes o empleados (LAMAS, 2014).

La prueba de rendimiento está diseñada para probar el rendimiento del *software* en tiempo de ejecución dentro del contexto de un sistema integrado.

Para las pruebas de carga se somete al sistema a una determinada carga de trabajo variable, mientras que el propio sistema en prueba permanece invariable. Es decir se simula la carga de trabajo promedio y con picos que pueden ocurrir dentro de tolerancias operacionales variables. Por su parte las pruebas de estrés están enfocadas a evaluar como el sistema responde bajo condiciones como pueden ser insuficiente memoria, extrema sobrecarga, servicios y *hardware* no disponibles entre otras condiciones anormales que pudieran presentarse en el sistema dado.

La herramienta utilizada para realizar las pruebas de rendimiento fue JMeter en su versión 2.9. Esta herramienta está diseñada para realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones Web. Permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento. Tiene la forma de generar un caso de prueba a través de una navegación de usuario.

El entorno de trabajo en el que fueron realizadas las pruebas de rendimiento cumple con las siguientes características: 1 PC cliente con un procesador AMD E-300 a 1.3 GHz y 4 GB de memoria RAM, con el sistema operativo Windows 8.0.

El plan de pruebas fue diseñado a partir de las diferentes acciones que los usuarios pueden realizar al conectarse al sitio web *OpenCourseWare*. Para una muestra de 50 usuarios conectados concurrentemente, con un período de subida de 1 segundo (tiempo de espera de cada usuario para realizar una petición) la herramienta JMeter generó los siguientes reportes.

Informe Agregado										
Nombre: Informe Agregado										
Comentarios										
Escribir todos los datos a Archivo										
Nombre de archivo				Navegar...		<input type="checkbox"/> Escribir en Log Sólo Errores		Configurar		
Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec	
/drupal7bet...	60	14	15	32	0	94	0,00%	19,4/min	1653,1	▲
/drupal7bet...	30	352	375	608	0	764	0,00%	20,8/min	363,9	
/drupal7bet...	30	356	390	640	0	1076	0,00%	20,7/min	1493,5	
/drupal7bet...	30	294	374	593	0	671	0,00%	20,6/min	671,1	
/drupal7bet...	30	386	343	686	0	2730	0,00%	20,5/min	1723,8	
/drupal7bet...	60	10	15	16	0	47	0,00%	19,3/min	1983,0	
/drupal7bet...	90	17	15	31	0	390	0,00%	26,2/min	402,2	
/drupal7bet...	30	314	359	608	0	1061	0,00%	20,3/min	6317,3	
/drupal7bet...	60	28	15	47	0	453	0,00%	19,3/min	662,9	
/drupal7bet...	60	27	15	46	0	843	0,00%	19,5/min	180,3	
/drupal7bet...	30	289	344	624	0	733	0,00%	20,2/min	959,0	
/drupal7bet...	30	274	374	500	0	733	0,00%	20,1/min	417,4	
/drupal7bet...	30	270	343	530	0	531	0,00%	20,0/min	5034,2	
/drupal7bet...	30	253	343	468	0	609	0,00%	20,0/min	364,6	
/drupal7bet...	30	313	375	577	0	1420	0,00%	19,7/min	214,8	
/drupal7bet...	60	48	15	47	0	1981	0,00%	19,4/min	72,7	
/drupal7bet...	60	13	0	31	0	250	0,00%	19,4/min	298,8	
/drupal7bet...	60	16	15	31	0	249	0,00%	19,4/min	184,8	
/safebrowsi...	30	7634	8034	13166	16	13166	100,00%	17,2/min	366,9	
/drupal7bet...	30	62280	63961	84460	38423	87143	100,00%	7,1/min	2475,2	≡
/drupal7bet...	30	29	15	94	0	312	0,00%	15,2/min	422,5	
/drupal7bet...	30	43834	47628	65646	16911	68626	0,00%	10,0/min	4186,6	
TOTAL	6510	1229	16	499	0	87143	0,00 %	18,4/sec	190403,5	▼

Figura 23: Reporte resumen generado por la herramienta JMeter.

A partir del resumen generado por la herramienta JMeter se puede observar que para un total de 6510 muestras realizadas al sistema alcanzó un rendimiento de 18,4 peticiones por segundo, con un porcentaje de 0.00% de errores para cada petición realizada. A partir de este resultado se considera que el sistema posee parámetros de rendimiento que aseguran las rápidas respuestas del sistema y la eficiencia del mismo.

Resultados de la validación del sistema.

Con esta validación se puede concluir que:

- La propuesta va a mejorar el proceso de recuperación de información que se realiza en la universidad.
- Los materiales digitales se organizan en una estructura de cursos que pertenecen a una categoría determinada respondiendo a las necesidades del cliente y permitiendo una eficiente presentación de la información requerida.
- El sistema obtenido automatiza los procesos de gestión de los materiales digitales dispersos bajo un enfoque editorial y con fines formativos y didácticos, contribuyendo con la integridad de los datos, demostrado los resultados satisfactorios obtenidos con las pruebas realizadas.

Conclusiones

Con el desarrollo de la solución propuesta y el alcance del resultado final se ha determinado que los objetivos específicos fueron cumplidos satisfactoriamente arribándose a las siguientes conclusiones:

- La investigación realizada permitió profundizar sobre el desarrollo de sitios web *OpenCourseWare*, distinguiendo los principales servicios que brindan y sus características generales, permitiéndose con esto, la determinación de las herramientas, metodología y tecnologías utilizadas en el desarrollo de la solución.
- La especificación de requisitos facilitó el análisis y diseño de las funcionalidades desarrolladas en el sitio web.
- La arquitectura seguida en la implementación permite incorporar futuras nuevas funcionalidades.
- El CMS Drupal constituye una herramienta muy potente en el desarrollo de aplicaciones web y posee altas prestaciones para la correcta implementación de un sitio web *OpenCourseWare*.
- Las pruebas de *software* efectuadas al producto permitieron identificar y corregir las no conformidades mejorando así la calidad del portal desarrollado.

Recomendaciones

- Generar un perfil de instalación de Drupal que incluya las funcionalidades del *OpenCourseWare* para facilitar su generalización.
- Diseñar una plantilla para la interfaz de usuario con un diseño adaptable
- Desplegar el sistema para su validación en un entorno real.

Bibliografía referenciada

1. AFFILIATES, O. A. O. I. (2011). *MySQL* [Consultado el: marzo de 2014]. (MySQL Reference Manual). Disponible en: <https://dev.mysql.com/doc/refman/5.0/es/>.
2. ALMEIRA, A. S. (2011). *Arquitectura de Software: Estilos y Patrones*. Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco, 2011.
3. ÁLVAREZ, M. A. (2010). *Desarrollo Web.com* de 2014]. (Manual de CSS 3). Disponible en: <http://www.desarrolloweb.com/manuales/css3.html>.
4. ANDALUCÍA, J. D. (2013). *Convenio de codificación específico para Drupal* España: 2013, Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/libro-pautas/124>.
5. ARA, I. (2013). *Asociación profesional de Ingenieros e Ingenieros técnicos en Informática de Cantabria* Cantabria: [Consultado el: marzo de 2014]. Disponible en: <http://www.ainitican.org/asamblea-y-charla-ainitican-17-de-diciembre-de-2013-en-la-facultad-de-ciencias/>. ISBN
6. CARO, P. (2012). *Google sites* de 2014]. (Metodología de gestión de requerimientos.). Disponible en: <https://sites.google.com/site/metodologiareq/capitulo-ii>.
7. COMUNICACIÓN, D. D. (2012). *Universitat Pompeu Fabra* Barcelona: [Consultado el: febrero de 2014]. Disponible en: http://www.upf.edu/decom/pdf/1213/doctorat_99_2011/PresentacionDigiDoc_SeminarioDoctorado-2012.pdf.
8. CÓRDOBA, O. O. D. L. U. N. D. (2013). *OpenCourseware de la Universidad Nacional de Córdoba* Argentina: Disponible en: <http://www.ocw.unc.edu.ar/>.
9. CSIC, E. (2009). *Especiales E-revistas* Disponible en: http://www.erevistas.csic.es/especial_revistas/revistas13.htm.
10. DRUPAL, C. H. D. (2011). *Arquitectura de Drupal*. 2011, vol. 555 x 268, Disponible en: <http://www.cursosdrupal.com/content/arquitectura>.
11. EGUÍLUZ, J. P. (2009). *Introducción a JavaScript*. 2009.
12. FRAMEWORK, P. E. P. (2013). *Metodología OpenUP*. publicado el: 2013 de 2013, última actualización: 2013. vol. 320x250, Disponible en: <http://srinclan.wordpress.com/2009/04/15/metodologias-agiles-parte-1/>.
13. FOUNDATION, P. (2012). *Plone.org* [Consultado el: febrero de 2014]. Disponible en: <http://plone.org/>.
14. GAMMA, E. (2005). *Patrones de diseño*. 2005.
15. GARCÍA, P. (2013). *Netbeans IDE* de 2014]. (5 Features You Will Love About NetBeans). Disponible en: <https://netbeans.org/>.
16. GIL, M. (2013). *Avance y perspectiva*. 2013, vol. 5, nº 1, p. 1-7. Disponible en:

- <http://avanceyperspectiva.cinvestav.mx/wp-content/uploads/2013/07/Acceso-abierto-origen-y-destino-Marti-Gil.pdf>.
17. GONZÁLEZ, L. E. M. (2014). *Metodología OpenUP*. 2014,
 18. GUZMÁN, C. L. (2006). *3R - Red de repositorios universitarios de recursos digitales. Primer informe técnico*. Macroproyecto Tecnologías para la Universidad de la Información y la Computación (UNAM). 2006
 19. KABIR, M. J. (2010). *La Biblia. Servidor Apache 2*. 2010,
 20. KUIL, A. V. D. (2004). "The Dawning of the Dutch Network of Digital Academic REpositories (DARE): A Shared Experience". *Ariadne Issue 41*, 2004, Disponible en: <http://www.ariadne.ac.uk/issue41/vanderkuil/>.
 21. LAMAS, J. (2014). *Pruebas de rendimiento*. Berlín, Alemania: 2014, vol. 2014, Disponible en: <http://www.globetesting.com/pruebas-de-rendimiento/>.
 22. LAPUENTE, M. J. L. (2013). *Hipertexto: El Nuevo concepto de documento en la cultura de la imagen*. Tesis Doctoral, Facultad de Ciencias de la Información. Dpto. de Biblioteconomía y Documentación. Universidad Complutense de Madrid. , 2013.
 23. LARMAN, C. (2003). *UML y Patrones*. Prentice Hall, 2003. Modelo del dominio.
 24. LEÓN, U. D. (2012). *OpenCourseWare de la Universidad de León* [Consultado el: marzo de 2014]. Disponible en: <http://ocw.unileon.es/>.
 25. MADRID, O. O. D. L. U. C. I. D. (2014). *OpenCourseWare de la Universidad Carlos III de Madrid* España: [Consultado el: febrero de 2014]. Disponible en: <http://ocw.uc3m.es/>.
 26. MARTÍN, R. C. (2009). *Principios y patrones de diseño*. [Consultado el: abril de 2014]. Disponible en: http://www.dsi.fceia.unr.edu.ar/downloads/informatica/info_III/patrones.ppt.
 27. MAYOR, A. C. (2013). *Centro de Comunicación y Pedagogía* [Consultado el: febrero de 2014]. Disponible en: <http://www.centrocp.com/cms-lms-y-lcms-definicion-y-diferencias/>.
 28. MÉXICO, U. A. D. E. D. (2013). *Red de Revistas Científicas de América Latina y el Caribe, España y Portugal* México: [Consultado el: marzo de 2014]. Disponible en: <http://www.redalyc.org/revista.oa?id=924>.
 29. MOLINA, J. L. P. (2013). *OpenCourseWare de la Universidad Politécnica de Cartagena*. Cartagena: [Consultado el: marzo de 2014]. Disponible en: <http://ocw.bib.upct.es/>.
 30. MUSEUM, C. H. (2014). *Computer History Museum* Última actualización: 2014. de 2014]. Disponible en: <http://www.computerhistory.org>.
 31. NOBREGA, M. D. (2013). *Herramientas CASE: Rational Rose*. Blogía, 2013, vol. 2014, Disponible en: [urso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php](http://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php).
 32. OVERVIEW, V. P. C. (2013). *Visual Paradigm* de 2014]. (Software design tools for agile software development.). Disponible en: <http://www.visual-paradigm.com/features/>.

33. PÉREZ, M. O. L. (2011). *TICs en América Latina* América Latina: [Consultado el: enero de 2014]. Disponible en: <http://www.tics.org.ar/home/index.php/noticias-destacadas-2/157-definicion-de-tics>.
34. POSTGRESQL (2012). 2012, vol. 2014, p. Disponible en: <http://postgresql-dbms.blogspot.com/p/limitaciones-puntos-de-recuperacion.html>.
35. PRESSMAN, R. S. (2010). *Software Engineering: A practitioner's approach*. New York: McGraw-Hill, 2010, vol. 7, ISBN 978-0-07-337597-7.
36. RAMÍREZ (2009). *Wordpress.com* México: (IDE, Tipos de Datos, Lenguajes de Programacion y Paradigmas de Programacion.). Disponible en: <http://cramirezr.wordpress.com/2009/10/20/ide-tipos-de-datos-lenguajes-de-programacion-y-paradigmas-de-programacion/>.
37. SALAMANCA, U. D. (2014). *Bibliotecas Salamanca*: [Consultado el: febrero de 2014]. Disponible en: <http://bibliotecas.usal.es/?q=acceso-abierto>.
38. SOCIALES, C. L. D. C. (2010). *Clacso.org* Buenos Aires, Argentina: [Consultado el: febrero de 2014]. Disponible en: http://www.clacso.org.ar/area_academica/2b3.php.
39. TARRAGÓ, N. S. (2007). *El sector editorial contemporáneo y las competencias profesionales*. Cuba: 2007, ISBN 2307 2113.
40. ULTIMATE, V. S. (2013). *Developer Network* de 2014]. (Diagramas de componentes de UML: Referencia). Disponible en: <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
41. UNIVERSIA, F. (2014). *OpenCourseware Universia* Madrid, España: de 2014]. Disponible en: <http://ocw.universia.net/es/>.
42. VALKENBURG, I. W. V. (2011). *OpenCourseWare and Standards*. 2011,
43. VÁZQUEZ, J. A. G. (2003). *Catálogo de libros, novedades, información y noticias de Editorial Anaya Multimedia*. . Editado por: Multimedia., E. A. Madrid: 2003, ISBN 84-415-1525-5.
44. VERACRUZANA, U. (2014). *Sitio Web de la Universidad Veracruzana* [Consultado el: marzo de 2014]. (Universo, el periódico de los universitarios). Disponible en: www.uv.mx/universo/486/infgral/infgral_15.html.
45. WEB, A. (2014). *Analítica Web* Disponible en: <http://latindesigner.net/algunas-caracteristicas-de-html5-que-usted-debe-saber/>.

Bibliografía consultada

1. AFFILIATES, O. A. O. I. (2011). *MySQL* [Consultado el: marzo de 2014]. (MySQL Reference Manual).
Disponible en: <https://dev.mysql.com/doc/refman/5.0/es/>.
2. ALMEIRA, A. S. (2011). *Arquitectura de Software: Estilos y Patrones*. Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco, 2011.
3. ÁLVAREZ, M. A. (2010). *Desarrollo Web.com* de 2014]. (Manual de CSS 3). Disponible en: <http://www.desarrolloweb.com/manuales/css3.html>.
4. ANDALUCÍA, J. D. (2013). *Convenio de codificación específico para Drupal España: 2013*, Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/libro-pautas/124>.
5. ARA, I. (2013). *Asociación profesional de Ingenieros e Ingenieros técnicos en Informática de Cantabria* Cantabria: [Consultado el: marzo de 2014]. Disponible en: <http://www.ainitican.org/asamblea-y-charla-ainitican-17-de-diciembre-de-2013-en-la-facultad-de-ciencias/>. ISBN
6. ARECHAVALA, Y. G. (2001). *Calidad del software(I)*. 2001, 1-60 p.
7. CARO, P. (2012). *Google sites* de 2014]. (Metodología de gestión de requerimientos.). Disponible en: <https://sites.google.com/site/metodologiareq/capitulo-ii>.
8. COMUNICACIÓN, D. D. (2012). *Universitat Pompeu Fabra* Barcelona: [Consultado el: febrero de 2014].
Disponible en: http://www.upf.edu/decom/_pdf/1213/doctorat_99_2011/PresentacionDigiDoc_SeminarioDoctorado-2012.pdf.
9. CÓRDOBA, O. O. D. L. U. N. D. (2013). *OpenCourseware de la Universidad Nacional de Córdoba* Argentina: Disponible en: <http://www.ocw.unc.edu.ar/>.
10. CSIC, E. (2009). *Especiales E-revistas* Disponible en: http://www.erevistas.csic.es/especial_revistas/revistas13.htm.
11. DRUPAL, C. H. D. (2011). *Arquitectura de Drupal*. 2011, vol. 555 x 268, Disponible en: <http://www.cursosdrupal.com/content/arquitectura>.
12. EGUÍLUZ, J. P. (2009). *Introducción a JavaScript*. 2009.
13. FRAMEWORK, P. E. P. (2013). *Metodología OpenUP*. publicado el: 2013 de 2013, última actualización: 2013. vol. 320x250, Disponible en: <http://srinlan.wordpress.com/2009/04/15/metodologias-agiles-parte-1/>.
14. FOUNDATION, P. (2012). *Plone.org* [Consultado el: febrero de 2014]. Disponible en: <http://plone.org/>.
15. GAMMA, E. (2005). *Patrones de diseño*. 2005.
16. GARCÍA, P. (2013). *Netbeans IDE* de 2014]. (5 Features You Will Love About NetBeans). Disponible en: <https://netbeans.org/>.
17. GIL, M. (2013). *Avance y perspectiva*. 2013, vol. 5, nº 1, p. 1-7. Disponible en: <http://avanceyperspectiva.cinvestav.mx/wp-content/uploads/2013/07/Acceso-abierto-origen-y-destino-Marti-Gil.pdf>.

18. GONZÁLEZ, L. E. M. (2014). *Metodología OpenUP*. 2014,
19. GUZMÁN, C. L. (2006). *3R - Red de repositorios universitarios de recursos digitales. Primer informe técnico*. Macroproyecto Tecnologías para la Universidad de la Información y la Computación (UNAM). 2006
20. HERRADÓN, R. (2012). *Objetivos, evolución y perspectivas del OpenCourseWare UPM*. Universidad Politécnica de Madrid. 2012
21. KABIR, M. J. (2010). *La Biblia. Servidor Apache 2*. 2010,
22. KUIL, A. V. D. (2004). "The Dawning of the Dutch Network of Digital Academic REpositories (DARE): A Shared Experience". *Ariadne Issue 41*, 2004, Disponible en: <http://www.ariadne.ac.uk/issue41/vanderkuil/>.
23. LAMANCHA, B. P. (2007). Gestión de las Pruebas Funcionales. *Actas de Talleres de Ingeniería del Software y Bases de Dato*, 2007, vol. 1, nº ISSN 1988-3455.
24. LAMAS, J. (2014). *Pruebas de rendimiento*. Berlín, Alemania: 2014, vol. 2014, Disponible en: <http://www.globetesting.com/pruebas-de-rendimiento/>.
25. LAPUENTE, M. J. L. (2013). *Hipertexto: El Nuevo concepto de documento en la cultura de la imagen*. . Tesis Doctoral, Facultad de Ciencias de la Información. Dpto. de Biblioteconomía y Documentación. Universidad Complutense de Madrid. , 2013.
26. LARMAN, C. (2003). *UML y Patrones*. Prentice Hall, 2003. Modelo del dominio.
27. LEÓN, U. D. (2012). *OpenCourseWare de la Universidad de León* [Consultado el: marzo de 2014]. Disponible en: <http://ocw.unileon.es/>.
28. MADRID, O. O. D. L. U. C. I. D. (2014). *OpenCourseWare de la Universidad Carlos III de Madrid España*: [Consultado el: febrero de 2014]. Disponible en: <http://ocw.uc3m.es/>.
29. MARTÍN, R. C. (2009). *Principios y patrones de diseño*. [Consultado el: abril de 2014]. Disponible en: http://www.dsi.fceia.unr.edu.ar/downloads/informatica/info_III/patrones.ppt.
30. MAYOR, A. C. (2013). *Centro de Comunicación y Pedagogía* [Consultado el: febrero de 2014]. Disponible en: <http://www.centrocp.com/cms-lms-y-lcms-definicion-y-diferencias/>.
31. MÉXICO, U. A. D. E. D. (2013). *Red de Revistas Científicas de América Latina y el Caribe, España y Portugal* México: [Consultado el: marzo de 2014]. Disponible en: <http://www.redalyc.org/revista.oa?id=924>.
32. MOLINA, J. L. P. (2013). *OpenCourseWare de la Universidad Politécnica de Cartagena*. Cartagena: [Consultado el: marzo de 2014]. Disponible en: <http://ocw.bib.upct.es/>.
33. MUSEUM, C. H. (2014). *Computer History Museum* Última actualización: 2014. de 2014]. Disponible en: <http://www.computerhistory.org>.
34. NOBREGA, M. D. (2013). *Herramientas CASE: Rational Rose*. Blogía, 2013, vol. 2014, Disponible en: [urso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php](http://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php).

35. OVERVIEW, V. P. C. (2013). *Visual Paradigm* de 2014]. (Software design tools for agile software development.). Disponible en: <http://www.visual-paradigm.com/features/>.
36. PÉREZ, M. O. L. (2011). *TICs en América Latina* América Latina: [Consultado el: enero de 2014]. Disponible en: <http://www.tics.org.ar/home/index.php/noticias-destacadas-2/157-definicion-de-tics>.
37. POSTGRESQL (2012). 2012, vol. 2014, p. Disponible en: <http://postgresql-dbms.blogspot.com/p/limitaciones-puntos-de-recuperacion.html>.
38. PRESSMAN, R. S. (2010). *Software Engineering: A practitioner's approach*. New York: McGraw-Hil, 2010, vol. 7, ISBN 978-0-07-337597-7.
39. RAMÍREZ (2009). *Wordpress.com* México: (IDE, Tipos de Datos, Lenguajes de Programacion y Paradigmas de Programacion.). Disponible en: <http://cramirezr.wordpress.com/2009/10/20/ide-tipos-de-datos-lenguajes-de-programacion-y-paradigmas-de-programacion/>.
40. SALAMANCA, U. D. (2014). *Bibliotecas Salamanca*: [Consultado el: febrero de 2014]. Disponible en: <http://bibliotecas.usal.es/?q=acceso-abierto>.
41. SOCIALES, C. L. D. C. (2010). *Clacso.org* Buenos Aires, Argentina: [Consultado el: febrero de 2014]. Disponible en: http://www.clacso.org.ar/area_academica/2b3.php.
42. SOMMERVILLE, I. (2005). *Ingeniería del Software (Séptima edición)*. Madrid, España: publicado el: 2014 de 2005, última actualización: 2014. ISBN 84-7829-074-5.
43. TARRAGÓ, N. S. (2007). *El sector editorial contemporáneo y las competencias profesionales*. Cuba: 2007, ISBN 2307 2113.
44. ULTIMATE, V. S. (2013). *Developer Network* de 2014]. (Diagramas de componentes de UML: Referencia). Disponible en: <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
45. UNIVERSIA, F. (2014). *OpenCourseware Universia* Madrid, España: de 2014]. Disponible en: <http://ocw.universia.net/es/>.
46. URIARTE, J. R. (2007). *OpenCourseWare: una alternativa para la publicación en abierto de contenidos educativos*. Universidad del País Vasco. 2007
47. VALKENBURG, I. W. V. (2011). *OpenCourseWare and Standards*. 2011,
48. VÁZQUEZ, J. A. G. (2003). *Catálogo de libros, novedades, información y noticias de Editorial Anaya Multimedia*. . Editado por: Multimedia., E. A. Madrid: 2003, ISBN 84-415-1525-5.
49. VERACRUZANA, U. (2014). *Sitio Web de la Universidad Veracruzana* [Consultado el: marzo de 2014]. (Universo, el periódico de los universitarios). Disponible en: www.uv.mx/universo/486/infgral/infgral_15.html.
50. WEB, A. (2014). *Analítica Web* Disponible en: <http://latindesigner.net/algunas-caracteristicas-de-html5-que-usted-debe-saber/>.

Glosario de términos

AJAX: Java Script asíncrono y XML (AJAX acrónimo de *Asynchronous Java Script And XML*).

Apache: Servidor web de distribución libre y de código abierto.

CASE: *Computer Aided Software Engineering*, en español Ingeniería de *Software* Asistida por Computadoras, son herramientas las cuales se definen como un conjunto de programas que asisten a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un *software*.

CIDI: Centro de Ideoinformática.

CSS: Hojas de Estilos en Cascada (CSS del inglés *Cascading Style Sheets*).

CMS: Sistema de Gestión de Contenidos (CMS del inglés *Content Management Systems*).

GNU/GPL: Licencia Pública General de GNU (GNU/GPL del inglés *GNU General Public License*).

HTTP: Protocolo de Transferencia de Hipertexto (HTTP del inglés *Hypertext Transfer Protocol*).

OAI-PMH: Iniciativa de Archivos Abiertos - Protocolo para la Transmisión de Metadatos (OAI-PMH del inglés *Open Archives Initiative – Protocol for Metadata Harvesting*).

PHP: Procesador de Hipertextos (PHP del inglés *Hypertext Pre-processor*).

Anexos

Anexo 1:

Entrevista realizada:

La presente entrevista tiene como finalidad realizar el levantamiento de requisitos y lograr un mayor entendimiento de los problemas existentes en el Movimiento de Acceso Abierto consolidado dentro de la Universidad.

Entrevistado: _____

Cargo: _____ Fecha: _____

- ¿Cuáles serán los servicios que brindará la solución?
- ¿Por qué surge la necesidad de implementar un sitio web *OpenCourseWare* en la universidad de las Ciencias Informáticas? ¿En qué influiría? ¿Qué mejoraría?
- ¿Cuáles son los principales contenidos que se promocionarían en el sitio web?
- ¿Qué roles poseerán los usuarios que interactúen con el sitio?
- ¿Qué características generales debe tener la solución?

Anexo 2:

Guía de observación:

- Desarrollo e incremento del Modelo de Acceso Abierto a la documentación académica y científica apoyado por el desarrollo de las TIC.
- *OpenCourseWare* como parte del Movimiento e Acceso Abierto: estado actual y principales exponentes.
- Perspectivas de Acceso Abierto dentro de la universidad.

Anexo 3:

Prototipos

Figura 24: Prototipo de interfaz de usuario del caso de uso Gestionar Materiales.

Figura 25: Prototipo de interfaz de usuario del caso de uso Gestionar lista de cursos.

Anexo 4:

Especificación de casos de uso:

Actores	Profesor : (Inicia)
----------------	---------------------

Resumen	El CU se inicia cuando el Profesor va a crear materiales dentro de un curso determinado.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones		
Pos condiciones		
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Se dirige al curso al cual quiere agregarle materiales.	1.1 Muestra un formulario con los campos del curso entre ellos el campo Material con las opciones: Crear material y Editar.
2.	El profesor selecciona una de las opciones	2.1 Si selecciona la opción "Crear material" ir a la sección Crear material. 2.2 Si selecciona la opción "Editar" ir a la sección Editar.
Sección 1: Crear materiales		
	Actor	Sistema
3	El profesor elige la opción "Crear materiales"	3.1 El sistema muestra un formulario en el que solicita llenar los datos que posee el material: Materiales, Tipo de Material y Adjunto.
4	El profesor inserta los datos a llenar.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 El sistema verifica que el material no haya sido agregado anteriormente. 4.3 El sistema notifica al profesor que creó un material correctamente.
Flujos alternos		
	Actor	Sistema
4	a) El profesor procede a llenar los datos del formulario y da clic en la opción "Guardar".	b) El sistema muestra un mensaje "Material X se ha creado".
	c) El profesor procede a llenar los datos del formulario y da clic en la opción "Vista previa".	d) El sistema muestra una vista previa del material que se crea. e) El sistema va a la página donde se muestra el

		material ya creado. Se termina el caso de uso.
Sección 2: Editar materiales		
	Actor	Sistema
5.	El profesor elige la opción "Editar materiales"	El sistema muestra un listado con todos los materiales que han sido creados.
6.	El profesor selecciona el material que desea Editar.	6.1 El sistema muestra un formulario con todos los datos del material seleccionado para ser cambiados. 6.2 El sistema verifica si los datos cambiados son correctos.
Flujos alternos		
	Actor	Sistema
6	a) El profesor procede a editar los datos del formulario y da clic en la opción "Guardar".	b) El sistema muestra un mensaje "Material X ha sido actualizado".
	c) El profesor procede a llenar los datos del formulario y da clic en la opción "Vista previa".	d) El sistema muestra una vista previa del material editado.
	e) El profesor no procede a llenar los datos del formulario y da clic en la opción "Eliminar". g) El profesor afirma querer realizar la acción.	f) El sistema verifica que se desee realizar esta acción. h) El sistema muestra un mensaje "Material X ha sido eliminado" g) El sistema va a la página donde se muestra el material ya editado. Se termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	

Tabla 9: Especificación del caso de uso Gestionar materiales.

Actores	Administrador : (Inicia)
Resumen	El CU se inicia cuando el Administrador

Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Existen categorías a las cuales se ha suscrito al menos un usuario.	
Postcondiciones		
Flujo de eventos		
Flujo básico		
	Actor	Sistema
3.	Se dirige al menú de administración y da clic en Contenido/ Agregar contenido/ Boletín.	1.1 Muestra un formulario con los campos a llenar para crear el contenido Boletín.
4.	Inserta los datos a llenar.	2.1 Verifica que los datos entrados sean correctos. 2.2 Notifica al administrador que el contenido ha sido guardado correctamente. 2.3 Muestra una vista previa del boletín creado
Flujos alternos		
1ª El boletín ha sido creado correctamente.		
	Actor	Sistema
1.	a) El administrador da clic en la opción "Boletín".	b) Muestra un formulario con la opción de "Enviar boletín".
2.	c) Da clic en el botón Enviar.	d) Muestra un mensaje "El boletín ha sido enviado correctamente".
Relaciones	CU Incluidos	
	CU Extendidos	

Tabla 10: Especificación del caso de uso Gestionar boletín.

Actores	Editor: (Inicia) Adiciona, edita y elimina cursos de la lista de cursos.
Resumen	El CU se inicia cuando el Profesor va a crear un curso
Complejidad	Alta
Prioridad	Crítico
Precondiciones	El editor se encuentra autenticado y tienen que existir cursos creados.

Postcondiciones	Quedan listados los cursos.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
5.	Se dirige al menú Editar Curso y selecciona la opción Lista de cursos.	1.1 El sistema muestra el formulario Edición de cursos que contiene las opciones Cursos por aprobar, Cursos Aprobados y Pendientes a revisión.
6.	El editor elige una de las opciones.	2.1 Si selecciona la opción "Cursos por aprobar" ir a la sección Cursos por Aprobar.
		2.2 Si selecciona la opción "Cursos Aprobados" ir a la sección Cursos aprobados.
		2.3 Si selecciona la opción "Pendientes a revisión" ir a la sección Pendientes a Revisión.
Sección 1: Cursos por aprobar		
	Actor	Sistema
3.	Selecciona la opción "Cursos por aprobar".	3.1 Muestra los cursos que han sido creados con anterioridad.
Flujos alternos		
	Actor	Sistema
3.	a) Selecciona un curso y da clic en la opción "Aceptar".	b) El sistema muestra un mensaje "La solicitud de contrato ha sido cambiada a: Aprobada satisfactoriamente"
	d) Selecciona un curso y da clic en la opción "Cancelar".	e) El sistema muestra un mensaje "La solicitud de contrato ha sido cambiada a: Revisión satisfactoriamente"
Sección 2: Cursos Aprobados		
	Actor	Sistema
4	Selecciona la opción "Cursos Aprobados".	4.1 Muestra los cursos que han sido aprobados anteriormente.
Flujos alternos		
4	a) Selecciona un curso y da clic en la opción "Cancelar".	b) Muestra un formulario donde debe introducirse las razones de rechazo del curso.
	c) Introduce los datos al formulario y da clic en "Aceptar".	d) Muestra un mensaje "La solicitud de contrato ha sido cambiada a: Por aprobar satisfactoriamente".

	e) Introduce los datos al formulario y da clic en "Cancelar".	f) El sistema muestra los cursos contenidos en la lista. Se termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	

Tabla 11: Especificación del caso de uso Gestionar Lista de cursos.

Actores	Editor (Inicia)	
Resumen	El caso de uso se inicia cuando el autor accede al sistema y le permite Gestionar artículos.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones		
Postcondiciones	Quedan creados los artículos por el autor correspondiente.	
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
7.	El autor se dirige al menú principal y selecciona el submenú Gestionar artículos.	1.1 El sistema muestra varias opciones entre las que se encuentran: Crear artículos, Modificar artículos y Eliminar artículos, Listar artículos.
8.	El autor selecciona una de las opciones.	2.1 Si selecciona "Crear artículos" ver sección Crear artículos. 2.2 Si selecciona "Modificar artículos" ver sección Modificar artículos. 2.3 Si selecciona "Eliminar artículos" ver sección Eliminar artículos. 2.4 Si selecciona "Listar artículos" ver sección Listar artículos.
Sección 1: Crear artículos		
	Actor	Sistema
3	El autor selecciona la opción de	3.1 El sistema muestra un formulario en el que pide llenar los datos que posee el artículo: Título, Imágenes,

	“Crear artículos”.	Resumen y Cuerpo.
4	El autor inserta los datos a llenar.	4.1 El sistema verifica que todos los campos del formulario estén llenos. 4.2 El sistema verifica que el autor no haya insertado anteriormente. 4.3 El sistema notifica al autor que creó el artículo correctamente.
Flujos alternos		
	Actor	Sistema
3	a) El autor procede a llenar los datos del formulario y da clic en la opción “Enviar”.	b) El sistema muestra un mensaje "Los datos del formulario fueron llenados satisfactoriamente.
	c) El autor no procede a llenar los datos del formulario y da clic en la opción “Cancelar”.	d) El sistema muestra un mensaje "Los datos del formulario Crear artículos, no fueron llenados satisfactoriamente". e) El sistema va a la página donde están las opciones de menú. Se termina el caso de uso.
Sección 2: Modificar artículos		
	Actor	Sistema
5	El autor selecciona la opción de “Modificar artículos”.	5.1 El sistema muestra a todos las publicaciones creadas por el autor.
6	El autor selecciona el artículo que desea Modificar.	6.1 El sistema muestra un formulario con todos los artículos realizados. 6.2 El sistema verifica si los datos cambiados son correctos.
Flujos alternos		
	Actor	Sistema
6	a) El autor procede a modificar los datos del formulario y da clic en la opción “Enviar”.	b) El sistema muestra un mensaje "Los datos del formulario fueron llenados satisfactoriamente".
	c) El autor no procede a llenar los datos del formulario y da clic en la opción “Cancelar”.	d) El sistema muestra un mensaje "Los datos del formulario Modificar artículos, no fueron llenados satisfactoriamente". e) El sistema va a la página donde están las opciones de menú. Se termina el caso de uso.
Sección 3: Eliminar artículos		

	Actor	Sistema
7	El autor selecciona la opción de "Eliminar artículos".	7.1 El sistema muestra a todos los artículos creadas por el autor.
8	El autor marca el artículo que desea Eliminar.	8.1 El sistema elimina el artículo de la base de datos.
Flujos alternos		
	Actor	Sistema
8	a) El autor procede a eliminar los datos del formulario y da clic en la opción "Enviar".	b) El sistema muestra un mensaje "Los datos del formulario fueron eliminados satisfactoriamente.
	c) El autor no procede a eliminar los datos del formulario y da clic en la opción "Cancelar".	d) El sistema muestra un mensaje "Los datos del formulario Eliminar revista, no fueron eliminados satisfactoriamente". e) El sistema va a la página donde están las opciones de menú. Se termina el caso de uso.
Sección 4: Listar artículos		
	Actor	Sistema
9	El autor selecciona la opción de "Listar artículos".	9.1 El sistema muestra a todos los artículos creados por el autor.
Relaciones	CU Incluidos	
	CU Extendidos	

Tabla 12: Especificación del caso de uso Gestionar artículos.

Anexo 5:

Diagramas de clases del diseño:

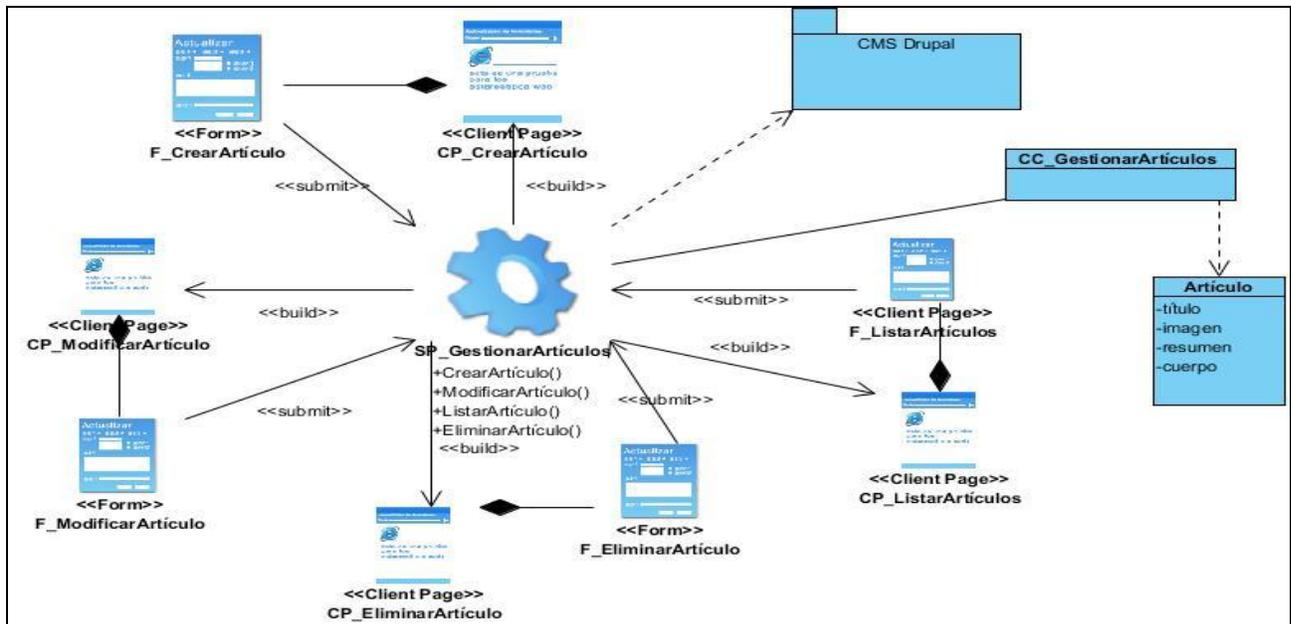


Figura 26: Diagrama de clases del diseño Gestionar artículos.

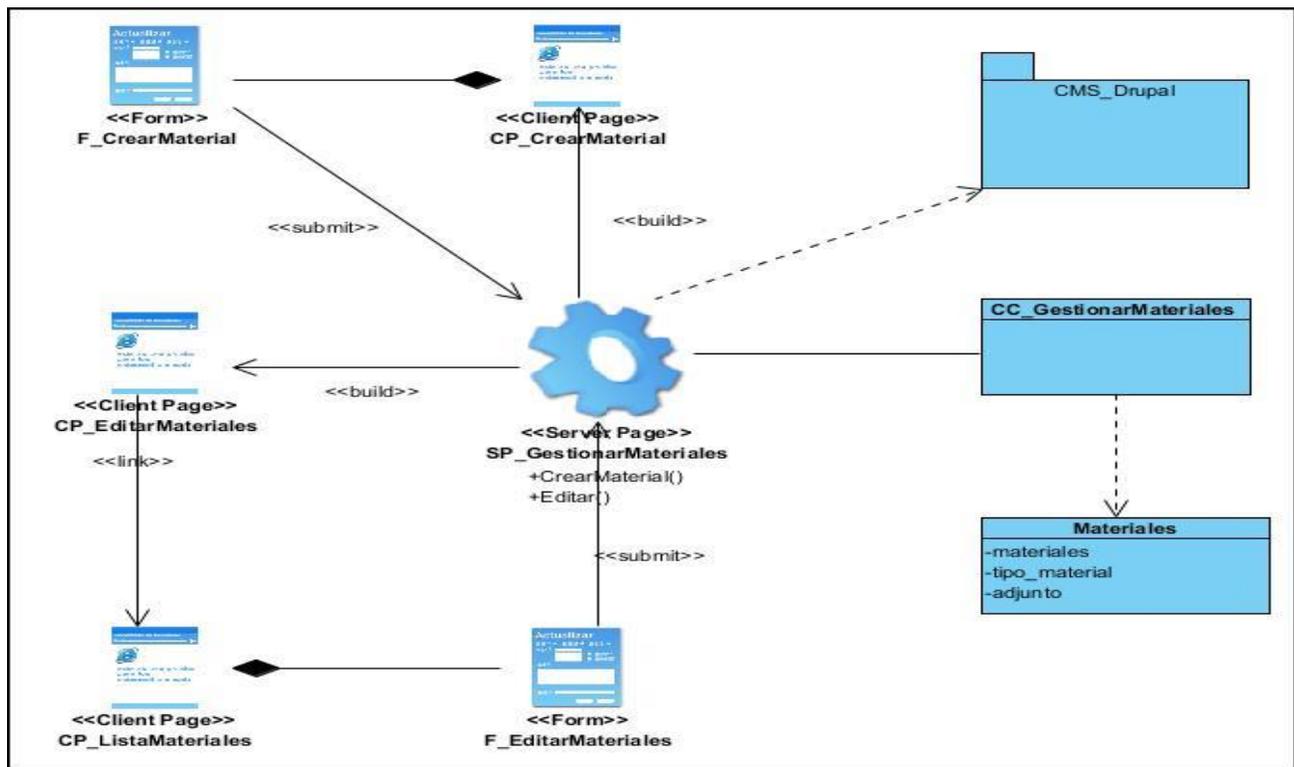


Figura 27: Diagrama de clases del diseño Gestionar materiales.

Anexo 6:

Diagramas de secuencia:

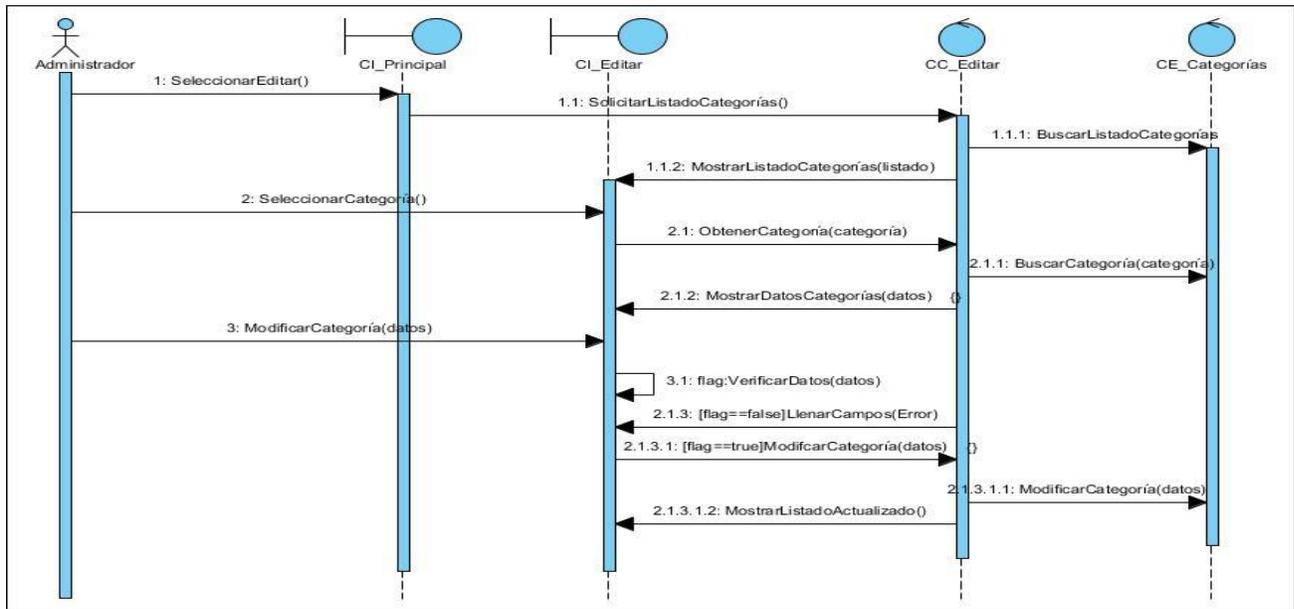


Figura 28: Diagrama de secuencia RF Editar Categoría

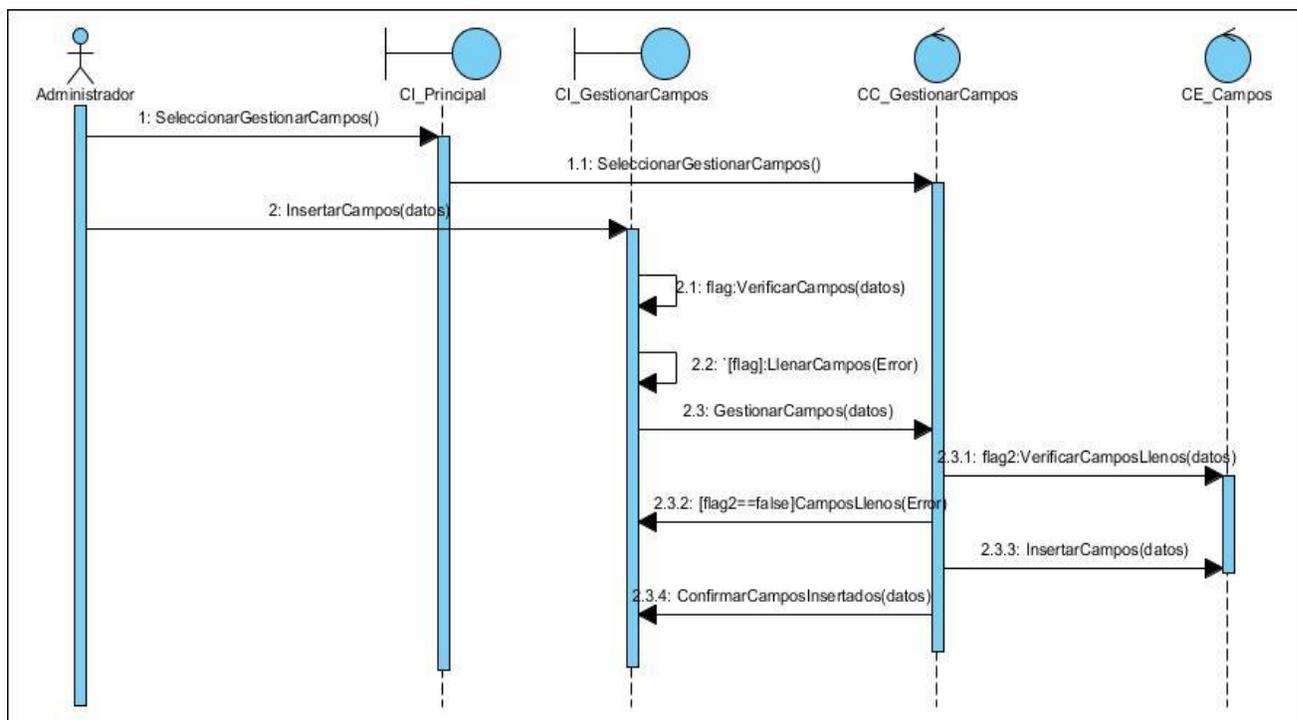


Figura 29: Diagrama de secuencia RF Gestionar campos.

Anexo 7:

Casos de prueba:

Escenario	Respuesta del sistema	Flujo central
EC 1.1 Crear material correctamente	El sistema guarda los datos correctamente y muestra un mensaje de confirmación "Material X se ha creado" Además muestra una vista previa del material.	<ol style="list-style-type: none"> 1. El usuario con permisos de profesor después de autenticado en el sistema accede al curso al cual le quiere agregar materiales, se dirige al campo Materiales del formulario que se muestra y selecciona Crear material.. 2. El sistema muestra un formulario en el que solicita llenar los datos que contiene el material. 3. El usuario con permisos de profesor inserta los datos y da clic en el botón Guardar.
EC 1.1.1 Crear material incorrectamente	El sistema no guarda los datos y muestra los mensajes de errores correspondientes: Para los campos obligatorios que estén vacíos se muestra el mensaje: El campo X es obligatorio.	<ol style="list-style-type: none"> 5. El sistema muestra el formulario para ingresar nuevamente los datos del nuevo material, y los mensajes de error correspondientes a los campos obligatorios vacíos. 6. El usuario con permisos de profesor ingresa los datos correctos y da clic al botón Guardar.
EC 1.2 Editar material.	El sistema guarda los datos correctamente, muestra una vista previa del material editado y muestra un mensaje "Material X ha sido actualizado."	<ol style="list-style-type: none"> 1. El usuario con permisos de profesor después de autenticado en el sistema accede al curso y dentro de este al campo Materiales y se dirige a la opción Editar. 2. El sistema muestra un listado con todos los materiales que han sido creados.. 3. El usuario con permisos de profesor selecciona el material que desea editar.. 4. El sistema muestra un formulario con todos los datos del material seleccionado para ser cambiados 5. El usuario con permisos de profesor modifica los datos y da clic en Guardar.

Tabla 13: Caso de prueba Gestionar materiales.

Escenario	Respuesta del sistema	Flujo central
EC 1.1 Autenticar usuario correctamente	El sistema permite el acceso del usuario a la sección.	<ol style="list-style-type: none"> 1. El usuario accede al sitio, luego escribe su usuario y contraseña y da clic en la opción Iniciar sesión. 2. El sistema verifica que los datos ingresados estén correctos y los campos obligatorios completos. 3. El sistema muestra el perfil del usuario y las opciones correspondientes.
EC 1.2 Autenticar	El sistema no permite el acceso del usuario a la sección,	<ol style="list-style-type: none"> 3. El sistema muestra

usuario incorrectamente	muestra los mensajes de error correspondientes y una vista nueva con las opciones de Iniciar sesión . En caso de que los campos estén vacíos el mensaje para cada campo es: "El campo X es obligatorio ". En caso que los datos sean incorrectos el mensaje es: Lo sentimos. No reconocemos el nombre de usuario o la contraseña. ¿Olvidó su contraseña?	nuevamente el formulario para el ingreso de usuario y contraseña. 4. El usuario ingresa los datos correctos y da clic en el botón Iniciar sesión . 5. El sistema verifica que los datos ingresados estén correctos y los campos obligatorios completos. 6. El sistema muestra el perfil del usuario y las opciones correspondientes.
-------------------------	---	--

Tabla 14: Caso de prueba Autenticar usuario.