

Universidad de las Ciencias Informáticas

Facultad 4



**Componentes para la gestión de páginas interactivas y
la integración de los recursos educativos.**

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

Autores:

Milena Tarrero Hernández.

Rubén Martínez Oliva.

Tutor:

Ing. Cindy Santos Salgado.

Ing. Osmany Casas Benítez.

Cotutores:

Ing. Isyed Rodríguez Trujillo.

La Habana, 2011

“Año 53 de la Revolución”

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente el ___ días del mes de _____ del año 2011.

Milena Tarrero Hernández

Rubén Martínez Oliva

Firma del Autor

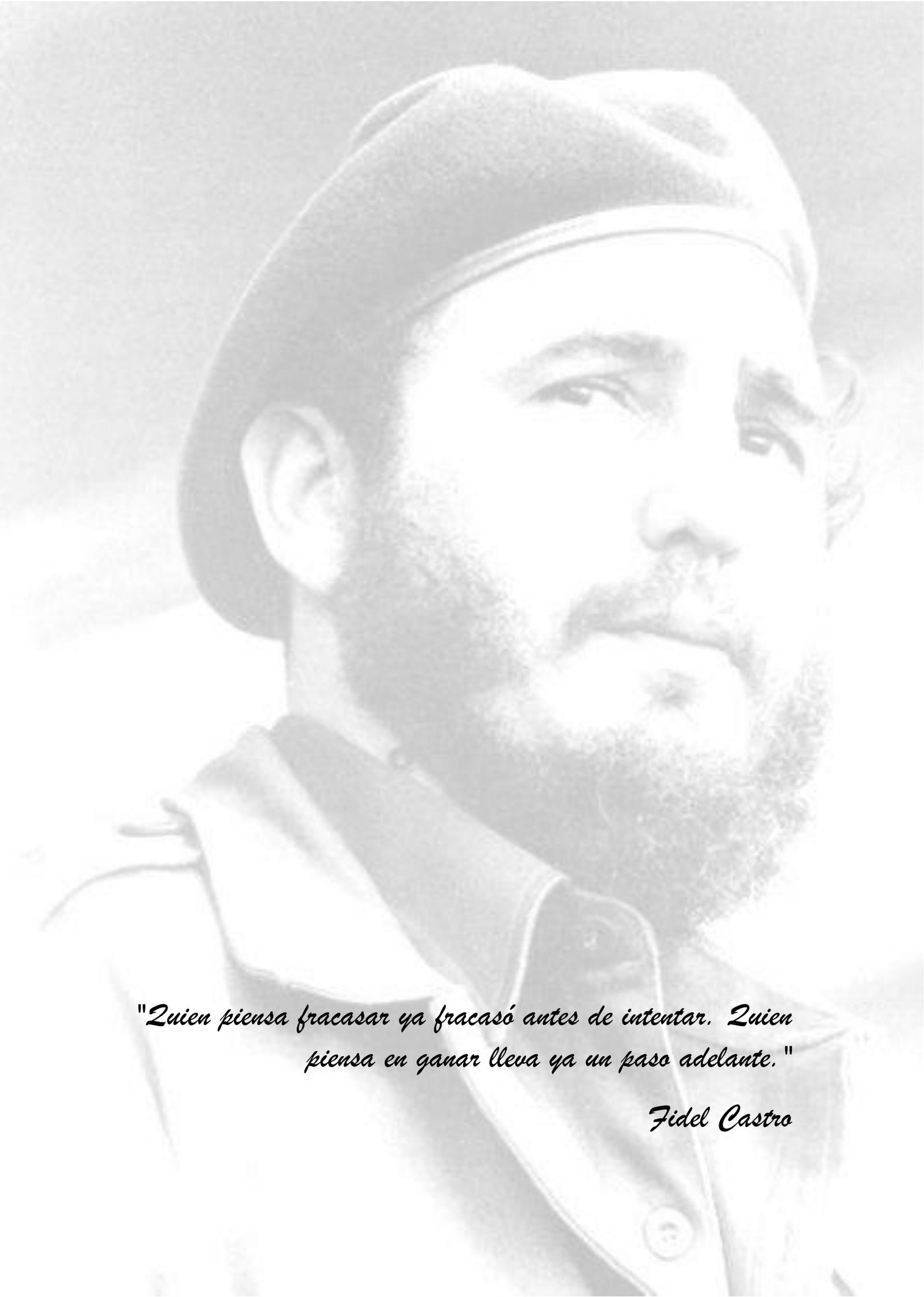
Firma del Autor

Ing. Cindy Santos Salgado

Ing. Osmany Casas Benítez

Firma del Tutor

Firma del Tutor



"Quien piensa fracasar ya fracasó antes de intentar. Quien piensa en ganar lleva ya un paso adelante."

Fidel Castro

DATOS DE CONTACTO

Nombre: Ing. Cindy Santos Salgado: Profesor graduado en el año 2010 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI). Pertenece al Centro de Tecnologías para la Formación (FORTES). Forma parte del proyecto Alfaomega de la Facultad 4.

Correo electrónico: csantos@uci.cu

Nombre: Ing. Osmany Casas Benítez: Profesor graduado en el año 2010 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI). Pertenece al Centro de Tecnologías para la Formación (FORTES). Forma parte del proyecto Alfaomega de la Facultad 4.

Correo electrónico: ocasas@uci.cu

De Milena

A la memoria de mi abuelo Rafael

A mi mamá, a mi papá y hermanos

A mi abuela Sonia

A mis amigos

De Rubén

A mi mamá, a mi papá, a mi hermana

A mi familia, a Yaneisi

A mis amigos

AGRADECIMIENTOS

A mi madre:

Por estar siempre ahí cuando la necesité, por ayudar a levantarme de los tropiezos de la vida, por estar siempre pendiente de mí, por quererme tal como soy.

A mi padre:

Por ser siempre mi ejemplo a seguir, por ser ese resabioso que tanto quiero y que siempre me guiado por el mejor camino, por darme el apoyo necesario a lo largo de la carrera.

A mi abuela Sonia:

Por ser más que eso, por ser una madre incondicional para mí, por su amor sin medida, por apoyarme en todo momento y por sobre todas las cosas confiar siempre en mí.

A Merlita:

Por estar ahí siempre para hacerme reír aun cuando mi ánimo no era el mejor, por aguantar mis malcriadeces y ser paciente conmigo.

A Yean:

Por ser esa persona incondicional que siempre he tenido al lado en las buenas y en las malas que me dio ánimo cuando lo necesité para levantarme después de una caída.

A mis tíos Maury, Mayi y Cacha:

Por motivarme siempre a seguir luchando y superándome cada día más, porque siempre confiaron en mí.

A Valdy:

Por estar a mi lado en estos últimos meses de la carrera sin importar la dificultad del momento apoyándome, por soportar con paciencia todas mis malacrianzas.

A mis tutores Cindy y Osmany:

Por su ayuda incondicional, por estar siempre ahí y guiarnos en todo el desarrollo de la tesis, por ayudarnos a calmar nuestros nervios en todas las exposiciones por las cuales pasamos.

A mis amigos Diana, Annia, Evelyn, Pedro González, Pedro Saéz, Suly, Lily, Daryl, Mora, Eduart por ser como una gran familia en estos años...el piketón de las fiestas...en fin a todos aquellos que estuvieron a mi lado de una manera u otra a lo largo de la carrera.

A mi compañero de tesis por su apoyo incondicional.

A mis compañeros de aula y proyecto en especial a May, Yola, Yula, Diosbel, Robe, Yaismel, Yuniel (El animal).

Al colectivo del Proyecto Alfaomega y a los profesores en general:

AGRADECIMIENTOS

Porque de todos aprendí algo, porque ayudaron en mi formación en el transcurso de estos años.

Quisiera agradecer a todas aquellas personas que de una forma u otra han tenido que ver con mi desarrollo como persona; a todas las amistades que hice en esta universidad, a mis amistades de Ciego, a mis profesores de la vocacional; sepan que lo que soy hoy como ser humano se debe en gran medida al aporte que hizo cada uno de ustedes en mi vida.

Milena

AGRADECIMIENTOS

A mis padres Miriam Oliva y Rubén Martínez:

Por el inmenso apoyo que me han brindado en toda mi vida. Por siempre estar presente cuando los he necesitado. Y por siempre haber contado con su cariño y dedicación.

A mi hermana:

Que siempre ha estado apoyándome y aconsejándome y que ha sido mi ejemplo a seguir, a Arian que es como mi hermano.

A mi novia Yaneisi:

Por haber estado en todo momento al lado mío apoyándome.

A mis abuelas, a mis tías y tíos, a mis primos, a toda mi familia y a todos los vecinos de la calle A.

A mis tutores Cindy y Osmany:

Por su ayuda incondicional.

A todos los profesores *que durante estos 5 años han contribuido con mi formación profesional.*

A todo el colectivo de Alfaomega, a mi compañera de tesis y a todos mis amigos.

Rubén

RESUMEN

Con el de cursar del tiempo Cuba se ha introducido en el mercado del software educativo. El país ha realizado software educativo para las diferentes enseñanzas, la Universidad de las Ciencias Informáticas constituye el pilar fundamental de la producción de software; la misma cuenta con diferentes proyectos productivos que realizan diversos productos para distintos sectores de la sociedad, entre ellos se encuentra la plataforma ZERA concebida para la gestión del aprendizaje basada en la concepción de hiperentornos de aprendizaje, los cuales son una fusión de diferentes software educativos sustentados en tecnología hipermedia, la misma tiene entre sus funcionalidades la gestión de los contenidos de aprendizaje de diferentes materias.

Teniendo en cuenta lo anterior el presente trabajo de diploma pretende ayudar a la gestión de los contenidos. Para ello se plantea la construcción de componentes que permitan la gestión de las páginas dentro de la plataforma. Se utilizará la metodología "Proceso Unificado de Desarrollo (RUP)" haciendo uso del framework de desarrollo Symfony. Se usan PHP y JavaScript como lenguajes de programación y se implementará el Modelo Vista Controlador como patrón arquitectónico a seguir. Como gestor de bases de datos se tendrá el PostgreSQL y como IDE de desarrollo el Net Beans. Estos componentes aportan como beneficio la posibilidad de gestionar las páginas de una manera organizada utilizando plantillas predefinidas por los usuarios para garantizar su estructuración.

Palabras Claves: páginas, plantillas, gestión de páginas, componentes.

INTRODUCCIÓN	1
FUNDAMENTACIÓN TEÓRICA	7
1.1. Introducción	7
1.2. Conceptos asociados al dominio del problema	7
1.2.1. Software educativo	7
1.2.2. Plataformas educativas	8
1.2.3. Sistema de gestión de aprendizaje.....	8
1.2.4. Sistema de gestión de contenidos para el aprendizaje.....	9
1.2.5. Estándar e-learning	9
1.2.6. Hiperentorno de aprendizaje	9
1.2.7. Sistema de gestión de contenidos	10
WordPress	11
Drupal	11
Joomla	12
1.3. Sistemas Similares	12
1.3.1. Claroline, plataforma educativa	12
1.3.2. Moodle	12
1.3.3. Plataforma ELEVEN	13
1.4. Editores de Texto	14
Editores WYSIWYG (What You See Is What You Get)	14
TinyMCE	14
CKEditor	15
1.5. Metodologías para el desarrollo del software	15
1.5.1. Rational Undefined Process (RUP)	15
1.5.2. Extreme Programming (XP).....	17
1.6. Lenguaje Unificado de Modelado (UML) como propuesta de modelación a utilizar en la solución	19
1.7. Lenguajes y herramientas seleccionados para el desarrollo	20
1.6.1. Del lado del cliente	20
XHTML.....	20
CSS	21

JavaScript.....	21
1.6.2. Del lado del servidor.....	21
PHP.....	21
Framework.....	22
Symfony.....	23
Jquery.....	24
ORM.....	24
Doctrine.....	25
1.8. Sistema Gestor de Base de Datos.....	26
1.7.1. PostgreSQL.....	26
1.9. Entorno de desarrollo integrado seleccionado.....	27
1.8.1. NetBeans IDE.....	27
1.10. Herramienta CASE para el modelado UML.....	27
1.9.1. Visual Paradigm.....	27
1.11. Herramientas para el Modelado de Prototipos.....	28
1.10.1. Balsamiq Mockups v1.6.58.....	28
1.10.2. Pencil Project.....	28
1.10.3. Dia.....	28
CARACTERÍSTICAS DEL SISTEMA.....	29
2.1. Introducción.....	29
2.2. Modelo de Dominio.....	29
2.2.1. Análisis de los conceptos del Dominio.....	29
2.2.2. Diagrama del Modelo de Dominio.....	31
2.2.3. Descripción del Sistema Propuesto.....	31
2.3. Requerimientos del software.....	31
2.3.1. Requerimientos Funcionales.....	32
2.3.2. Requerimientos No Funcionales.....	33
2.4. Modelo de Casos de Uso del Sistema.....	35
2.4.1. Descripción de los actores del sistema.....	35
2.4.2. Diagrama de actores del sistema.....	36
2.4.3. Diagrama de casos de uso del sistema.....	36

2.4.4.	Descripción de los casos de uso del sistema	37
ANÁLISIS Y DISEÑO DEL SISTEMA		41
3.1.	Introducción.....	41
3.2.	Modelo del análisis.....	41
3.2.1	Diagrama de clases del análisis	42
3.2.2	Diagramas de colaboración del análisis	42
3.3.	Patrón arquitectónico Modelo – Vista – Controlador en symfony	43
3.4.	Modelo de diseño	44
3.4.1.	Diagrama de clases del diseño.....	44
3.5.	Diseño de la base de datos	45
3.4.1.	Descripción de las tablas	45
3.6.	Conclusiones parciales	49
IMPLEMENTACIÓN Y PRUEBA.....		50
4.1	Introducción.....	50
4.2	Modelo de Implementación.....	50
4.2.1	Diagrama de Despliegue	50
4.2.2	Diagrama de Componentes.....	51
4.3	Pruebas de Software	54
4.3.1	Niveles de Prueba	54
4.3.2	Método de Prueba	55
4.3.2.1	Diseño de Casos de Prueba	55
4.4	Resultados Obtenidos.....	60
4.5	Conclusiones	61
CONCLUSIONES.....		63
BIBLIOGRAFÍA.....		65
GLOSARIO DE TÉRMINOS.....		67

INTRODUCCIÓN

El software educativo constituye una muestra fidedigna del impacto que han tenido las tecnologías de la información y las comunicaciones (TIC) en el ámbito educacional, facilitando la tarea de los educadores y brindando la posibilidad a todos de aprender y aumentar sus conocimientos en diferentes ramas de la ciencia.

La utilización de las TIC tiene grandes ventajas, tales como el enriquecimiento de la pedagogía al incorporar la tecnología de punta que revoluciona los métodos de enseñanza-aprendizaje, permite elevar la calidad del proceso docente-educativo y muestra la interdisciplinariedad de las asignaturas. Ofrece la posibilidad de comunicación interpersonal, para trabajo en equipo e intercambio de información e ideas, funciona como una herramienta para la gestión y el procesamiento de la información; constituye la unión de recursos interactivos que facilitan el aprendizaje y el desarrollo psicomotor y cognitivo del individuo.

El software desarrollado para la enseñanza permite la interactividad con los estudiantes, provee conocimientos que permiten la retroalimentación y a su vez la evaluación de lo aprendido; fomenta el desarrollo de habilidades a través de la ejercitación, brinda la posibilidad de simular procesos complejos; reduce el tiempo que se tiene para impartir gran cantidad de contenido facilitando a su vez el trabajo diferenciado, introduciendo al estudiante a un mundo nuevo constituido de medios computarizados con técnicas más avanzadas, que proveen al mismo de medios más completos y didácticos para facilitar su aprendizaje en un ambiente interactivo. Estas nuevas tecnologías puestas a su disposición no sustituyen la manera tradicional de impartir clases, sino que funcionan como un complemento, pero sí aportan facilidades a los docentes a la hora de utilizar medios que ayudan a los estudiantes a interiorizar mejor los contenidos y a su vez les proporciona una amplia documentación del tema.

En los últimos años los Sistemas de Gestión de Aprendizaje (por sus siglas en inglés *LMS, Learning Management Systems*), atraen la atención de las instituciones educacionales por las posibilidades que ofrecen en la gestión de actividades formativas, la creación de entornos virtuales de aprendizaje de manera sencilla y la posibilidad de ofertar cursos en modalidades semipresencial o no presencial.

Un LMS es una aplicación de software que se instala en un servidor, este tiene funciones que son fundamentales tales como la gestión de usuarios, recursos, así

como materiales y actividades de formación, administrar el acceso y hacer el seguimiento del proceso de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión, videoconferencias, entre otros. Este generalmente no incluye posibilidades de autoría (que permite crear sus propios contenidos), esta labor de crear los contenidos para los cursos se desarrolla mediante un Sistema de Gestión de Contenidos de Aprendizaje (conocido por sus siglas en inglés *LCMS, Learning Content Management Systems*).

El LCMS es un entorno multiusuario que permite que los desarrolladores puedan crear, almacenar, reutilizar, gestionar y distribuir contenidos de aprendizaje partiendo de un repositorio de objetos de aprendizaje; este sistema es ideal en el sector educativo debido a que combina las capacidades de gestión de cursos de un LMS con las capacidades de almacenamiento y creación de contenidos de un Sistema de Gestión de Contenidos (por sus siglas en inglés *CMS, Content Management Systems*), estos últimos facilitan la gestión de las webs.

Cuba no se ha quedado exenta respecto a su inserción en el mercado mundial del software, en el país se ha visto un avance en cuanto a la producción de software dedicados al sistema educacional, útiles para el aprendizaje en diversas enseñanzas asistidas por computadoras.

La Universidad de las Ciencias Informáticas (UCI) juega un determinante papel en el desarrollo de este tipo de software, vinculándose de manera directa en la informatización del proceso de enseñanza-aprendizaje, la misma cuenta con varios logros en el sector educativo ya que posee una amplia gama de proyectos que se dedican a la producción de software; una muestra de esto es el proyecto Alfaomega el cual se dedica al desarrollo de ZERA, una plataforma para la gestión del aprendizaje basada en la concepción de hiperentornos de aprendizaje, los cuales son una fusión de diferentes software educativos sustentados en tecnología hipermedia.

La plataforma ZERA está constituida por diferentes subsistemas que a su vez engloban diversos módulos que integrados conforman un ambiente sensible a las necesidades de los usuarios que hagan uso de la misma. Dicha plataforma debe contar con un módulo que se encargue de la edición de páginas.

En la actualidad existen editores que se encuentran limitados a la hora de asociar diferentes recursos interactivos a la página, estos lo hacen de manera muy variada

pues no todos insertan cualquier tipo de recursos, algunos permiten insertar texto, imágenes, contenidos, multimedia, gif's animados, elementos web (formularios, marcos, enlaces, botones) y posibilitan crear tablas; pero en otros casos no dan la posibilidad de asociarle a las páginas otros recursos que permita hacerle llegar al usuario de la forma más interactiva y amena posible la información, estos pueden ser simulaciones, videos, sonidos, curiosidades, palabras clave que contengan a su vez recursos asociados a ellas, entre otros.

Los editores de hoy no logran la mejor integración con otros módulos de la plataforma en la que están insertados; también poseen limitaciones en cuanto a la creación de interfaces líquidas que se visualicen en cualquier dispositivo electrónico diferente a un ordenador. Algunos poseen plantillas o formatos de páginas predeterminadas y en muchos casos a estas no se les puede cambiar su contenido.

En ZERA se tiene la necesidad de lograr una excelente integración con los demás módulos que la conforman, lograr el desarrollo de diferentes componentes para obtener la gestión de páginas interactivas y la integración de recursos educativos, los mismos brinden la posibilidad al usuario de crear sus propias plantillas de manera que se obtenga el formato deseado, además, a su vez separe el contenido y el diseño en HTML y CSS, lo que permita de una manera sencilla y fácil al usuario configurar el estilo de su página de la forma más interactiva posible. Surge entonces como **problema de investigación** la siguiente interrogante: ¿Cómo lograr la gestión de los contenidos de las materias y posibilitar la integración de ellos con los recursos interactivos en la plataforma para la gestión del aprendizaje ZERA?

Para dar solución al problema planteado se tiene como **objetivo general**: Desarrollar componentes que permitan gestionar los contenidos y posibilitar la integración de recursos interactivos a estos en la plataforma para la gestión del aprendizaje ZERA.

Para el desarrollo de la investigación se define como **objeto de estudio** el proceso de gestión de páginas y la asociación de recursos interactivos en el desarrollo de plataformas para la gestión del aprendizaje.

Se define como **campo de acción** el desarrollo de una herramienta para la gestión de páginas en la plataforma para la gestión del aprendizaje ZERA.

Objetivos específicos:

1. Realizar el estudio del estado del arte.
2. Definir los requerimientos funcionales necesarios para el desarrollo del módulo iPages dentro de la plataforma ZERA.
3. Realizar el análisis y diseño del módulo iPages.
4. Implementar el módulo iPages para la gestión de las páginas de los contenidos de las materias en la plataforma ZERA.
5. Realizar pruebas de integración del módulo iPages con la plataforma ZERA.

El trabajo defiende la idea que si se implementa un módulo para la gestión de las páginas que permitan la gestión de contenidos y que permita la asociación de recursos a estos, se favorecerá el trabajo de los docentes y se lograrán páginas lo más interactivas posibles.

Para todo el proceso de investigación y elaboración de este trabajo se tomará en cuenta la utilización de varios **métodos científicos de investigación** como:

En el nivel teórico:

Dentro de los métodos teóricos existentes se seleccionó para ser aplicado en la investigación el **histórico-lógico** con el propósito de conocer la evolución de los editores de páginas integrados en plataformas para la gestión del aprendizaje a raíz del auge de las TIC en el desarrollo de software para el sector educativo. Para conformar el marco teórico se utilizó el método **analítico-sintético** el cual brinda la posibilidad de estudiar y analizar la información disponible acerca de plataformas de gestión de aprendizaje. Se utilizó la **modelación** para realizar un diseño simplificado de la realidad a través de diagramas de clases, diagramas de flujo y diagramas de componentes, y de esta forma llevar a una vista más real el editor de páginas que se pensó.

En el nivel empírico:

Se empleó el método de la **observación** para obtener datos con respecto a la necesidad existente a lo largo del ciclo de desarrollo del proyecto. Se hizo uso del **análisis de fuentes de información** para examinar las distintas fuentes de información como internet, libros, revistas, entre otras.

Con el fin de satisfacer los objetivos específicos se proponen las siguientes **tareas de investigación**:

- Investigación sobre los distintos lenguajes y frameworks de desarrollo existente.
- Investigación sobre los principales CMS, haciendo énfasis en la gestión de páginas que realizan y en los editores de texto que utilizan.
- Investigación acerca de las posibles metodologías y herramientas a utilizar en el desarrollo.
- Elaboración del levantamiento de requisitos funcionales a tener en cuenta para el desarrollo del módulo.
- Definición de los casos de uso arquitectónicamente significativos.
- Análisis y diseño del módulo gestión de contenidos para la plataforma ZERA.
- Elaboración de la documentación correspondiente a los Flujos de Trabajo propuestos por la metodología a utilizar.
- Implementación de las funcionalidades descritas para el desarrollo del módulo iPages.
- Desarrollo de pruebas de funcionalidad a la herramienta obtenida.

El presente trabajo está estructurado en 4 capítulos:

Capítulo 1: Fundamentación Teórica

Contiene la fundamentación teórica del tema que soporta la investigación, abordando principalmente las tecnologías y lenguajes de programación que se utilizan en el desarrollo del módulo. Además, se exploran diferentes soluciones existentes semejantes al campo de acción para con ello obtener una medida de las posibles automatizaciones a realizar en el módulo iPages.

Capítulo 2: Características del sistema

Contiene el modelamiento del negocio y el levantamiento de requisitos; se obtienen los principales artefactos que generan estos flujos de trabajo propuestos por la metodología a utilizar.

Capítulo 3: Análisis y Diseño del sistema.

Se realiza el diseño del módulo, así como los diagramas de clases web, que brindan una visión clara del producto.

Capítulo 4: Implementación y prueba.

Los aspectos del desarrollo son tratados en este capítulo. Se describe cómo está implementado el módulo, presentando los diagramas de componentes y despliegue. También se describen las pruebas realizadas al módulo para comprobar su correcto funcionamiento.

FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se abordarán de manera sintetizada los elementos teóricos de los principales editores de páginas y soluciones propuestas a nivel mundial para la edición de páginas en plataformas educativas. Además, se abordará acerca de las principales tecnologías y lenguajes de programación que se utilizan en el desarrollo del módulo, explorando diferentes soluciones existentes semejantes al campo de acción para con ello adquirir conocimientos que permitan una mejor automatización a realizar en el módulo.

1.2. Conceptos asociados al dominio del problema

1.2.1. Software educativo

Se conoce como software educativo al destinado a la enseñanza y el aprendizaje individual o independiente y que además, permite el desarrollo de ciertas habilidades cognitivas. Para la creación de software educativo existe una amplia gama de enfoques ya sea atendiendo a los diferentes tipos de interacción que se tenga entre los actores del proceso de enseñanza-aprendizaje, como educador, aprendiz, conocimiento, computadora. Existen desde programas orientados al aprendizaje, sistemas orientados al aprendizaje hasta sistemas operativos destinados a la educación; un ejemplo de ello es una distribución de Linux orientada a la enseñanza. Programas para ordenadores creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y aprendizaje. Esta definición engloba todos los programas que han estado elaborados con fin didáctico, desde los tradicionales programas basados en los modelos conductistas de la enseñanza, los programas de Enseñanza Asistida por Ordenador (EAO), hasta los aún programas experimentales de Enseñanza Inteligente Asistida por Ordenador (EIAO), que utilizando técnicas propias del campo de los Sistemas Expertos y de la Inteligencia Artificial en general, pretenden imitar la labor tutorial personalizada que realizan los profesores y presentan modelos de representación del conocimiento en consonancia con los procesos cognitivos que desarrollan los alumnos [1].

A raíz del estudio y la observación de las ideas expuestas anteriormente referentes a conceptos se puede definir que el software educativo esencialmente está caracterizado por ser interactivo, a partir del empleo de recursos multimedia, ejercicios

y juegos instructivos que apoyan las funciones de evaluación y calificación. Posibilita el intercambio con los estudiantes evaluando lo aprendido y facilita el desarrollo de habilidades por medio de la evaluación.

1.2.2. Plataformas educativas

El uso de plataformas educativas para la administración de cursos proporciona la autonomía de producción y publicación en la red de recursos y contenidos por parte de los usuarios que son partícipes de la misma. El pedagogo, teniendo en cuenta los objetivos educativos, autónomamente y con la posibilidad de editar en cualquier momento basándose en alguno de los recursos dados por la plataforma tales como el programa del curso, contenidos o unidades temáticas expuestos en textos, hipertextos, presentaciones, animaciones, videos entre otros, actividades, bibliografía y evaluación, puede editar los contenidos a disposición de los estudiantes. De manera análoga, con otro nivel de autonomía, el estudiante puede acceder a los contenidos y al desarrollo de las actividades propuestas [2].

Las plataformas educativas posibilitan la estimulación de la idea de cooperación y de interacción, como aspectos centrales del proceso de aprendizaje y enseñanza, mediante el uso de herramientas colaborativas que favorecen la adquisición de aprendizajes significativos en los estudiantes y que al mismo tiempo afianzan en las prácticas docentes de enseñanza mediadas por las Tecnologías de la Información y la Comunicación (TIC) [2].

En resumen, las plataformas educativas favorecen sin dudas el proceso de aprendizaje de los estudiantes y tiene como factor clave que los alumnos pueden intercambiar entre ellos a través de los debates, foros, chats y constituyen un espacio más a dónde dirigirse en caso de que se les presente dudas en algún tema de investigación.

1.2.3. Sistema de gestión de aprendizaje

Los sistemas de gestión de aprendizaje (*Learning Management Systems* o *LMS*) son aplicaciones Web que proveen las funciones administrativas y de seguimiento necesario para posibilitar y controlar el acceso a los contenidos, implementar recursos de comunicaciones y llevar a cabo el seguimiento de quienes utilizan la herramienta. En general, los LMS facilitan la interacción entre los docentes y los estudiantes, aportan herramientas para la gestión de contenidos académicos y permiten el

seguimiento y la evaluación. Es decir, facilitan la “simulación” del modelo real en el mundo virtual, por lo que también se les conoce como Entorno Virtual de Aprendizaje (EVA) [2].

1.2.4. Sistema de gestión de contenidos para el aprendizaje

Los sistemas de gestión de contenidos para el aprendizaje (*Learning Content Management Systems* o *LCMS*) son LMS que permiten la gestión de contenidos. Entendiendo por gestión de contenidos el proceso que va desde la creación de un objeto de aprendizaje (OA), que es la unidad mínima de contenido, hasta su publicación y seguimiento. Estos fueron diseñados para generar la descripción de cada OA, buscar y localizar el OA requerido, proveer jerarquías para el almacenamiento y organización de un OA y ensamblar OA para estructurar cursos [2].

1.2.5. Estándar e-learning

E-learning se refiere al uso de las tecnologías basadas en Internet para distribuir y entregar soluciones que fomenten el conocimiento y desempeño. Rosenberg (2002) menciona que existen tres aspectos básicos en cuanto al e-learning tales como que está basado en redes informáticas, lo cual permite actualizar, almacenar, consultar, distribuir y colaborar en comunidades de aprendizaje; además está basado en Internet, para entregarse toda la información vía una computadora personal; también se enfoca en la manera más general del aprendizaje, yendo más allá de los paradigmas educativos [3].

Con todos los elementos dados se puede incorporar el e-learning en distintos sistemas educativos, pues con la aparición del mismo y el aprendizaje basado en Internet se han desarrollado diversos sistemas y plataformas que posibilitan la distribución de la información y a su vez la gestión de los diferentes cursos impartidos por diversos docentes, lo cual proporciona un ambiente fácil para conjuntar diferentes servicios y herramientas propias del mismo sentido [3].

1.2.6. Hiperentorno de aprendizaje

Un hiperentorno de aprendizaje se define como una modalidad informática que se sustenta en la tecnología hipermedia y en el que están presentes un conjunto de elementos representativos de diversas tipologías de software educativo [4].

Estos hiperentornos tienen en común la utilización de un ambiente informático lo más didáctico posible, integrando a estudiantes y profesores con las facilidades que

ofrecen los sistemas informáticos, esto proporciona condiciones de interacción local o a través de redes, así como posibilidades de acceso a recursos formativos, locales o distribuidos.

Los hiperentornos de enseñanza y aprendizaje deben estar comprendidos dentro de la lógica interna de los componentes del proceso de enseñanza-aprendizaje y esta relación e interacción con otros medios, así como los componentes restantes del proceso dentro del contexto educativo donde se desarrolla el mismo proporcionan una interacción de los diferentes sujetos que participan [4].

El uso de esta tecnología ha traído una serie de resultados favoreciendo el empleo del uso de métodos didácticos tales como la libertad de navegación que le da al estudiante la posibilidad de obtener información precisa en el orden que desea; se favorece el aprendizaje exploratorio de los estudiantes y se propicia que estos adquieran un papel activo mediante su interacción con el sistema; al navegar en un hiperentorno, es posible lograr la transferencia al estudiante de la red semántica implementada en el mismo; estos sistemas favorecen procesos de integración y contextualización de los conocimientos; la presentación de información multimedia resulta más enriquecedora y amena, lo que incrementa el uso y el atractivo del sistema para los estudiantes [4].

1.2.7. Sistema de gestión de contenidos

Un sistema de gestión de contenidos es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes.

Consiste en una interfaz que controla una o varias bases de datos donde se encuentra el contenido del sitio, este permite operar de manera independiente el contenido y el diseño, esto hace posible darle en cualquier momento un diseño diferente al sitio sin necesidad de dar formato al contenido nuevamente, permite además la fácil y a la vez controlada publicación en el sitio a varios editores. Ejemplo de lo antes mencionado son los editores que cargan el contenido al sistema y otro nivel superior llamado directorio que permite que estos contenidos sean aprobados y visibles a todo el público.

El uso de CMS y la tecnología enmarcados en la sociedad de la información, es un medio que permite al docente y al alumno acercar información útil y relevante para el

desarrollo de contenidos promoviendo la construcción de un espacio comunicacional adecuado para el proceso de formación académico. El docente puede valerse del potencial de estos soportes tecnológicos para orientar, acompañar y contener el proceso formativo del estudiante y del grupo.

WordPress

WordPress es un CMS desarrollado bajo el lenguaje PHP con el propósito de utilizar el gestor de base de datos MySQL y liberado bajo la licencia GPL. Por la facilidad que ofrece su uso, su licencia y su gran comunidad de desarrolladores y diseñadores alcanzó a convertirse en el CMS libre más utilizado en el 2009, según la editorial Packt el navegante Publishing [5].

Este establece tres estructuras organizativas fundamentales para la gestión de contenidos, tales como la categoría, la etiqueta y el artículo. Dentro de los artículos se encuentra toda la información a publicar y estos pueden ser agrupados por categorías y por etiquetas. Las categorías y las etiquetas se encuentran al mismo nivel, es decir, no se pueden etiquetar las categorías ni categorizar las etiquetas. La idea es facilitar la búsqueda de un artículo en específico agrupándolo en categorías y etiquetándolo [5].

WordPress establece una generación dinámica de páginas debido a que no es necesario reconstruir todas sus páginas cada vez que actualiza su bitácora o cambia algún detalle de la misma; todas las páginas se generan al hacer uso de la base de datos y las plantillas cada vez que es solicitada por un visor, esto significa que actualizar los datos o su diseño es tan rápido como sea posible, y el espacio utilizado en el servidor es mínimo [5].

Drupal

Drupal es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Se destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema. Es un excelente CMS para desarrollar aplicaciones web con una gran cantidad de plugins y temas visuales.

Todo el contenido en Drupal se maneja como un nodo, lo que permite trabajar con una serie de datos comunes entre ellos. Un nodo puede ser toda una página y una página

puede estar formada por varios nodos. El contenido está organizado a través de una taxonomía muy flexible que permite relaciones jerárquicas o lineales.

Joomla

Joomla es un sistema gestor de contenidos dinámicos que permite crear sitios web de alta interactividad, gran calidad, profesionalismo y eficiencia. La administración de Joomla está enteramente basada en la gestión de contenidos "en línea" y en "tiempo real". Este CMS está desarrollado con PHP para el gestor de base de datos MySQL bajo licencia GPL. El contenido está organizado en una estructura jerárquica bastante rígida que se divide en secciones, categorías y artículos. Todos los artículos pertenecen a una sola categoría y una categoría pertenece a una sección. Los artículos representan las páginas y pueden contener imágenes, videos, entre otros.

1.3. Sistemas Similares

1.3.1. Claroline, plataforma educativa

Claroline es una plataforma para la enseñanza virtual publicada bajo una licencia Open Source. Permite a los creadores construir eficaces cursos online y la gestión de actividades de aprendizaje y colaboración en la web. La plataforma está establecida alrededor del concepto de espacios relacionados con un curso y actividad pedagógica, cada espacio suministra una lista de herramientas que permite la creación de contenidos de aprendizaje y gestión de actividades de formación [6].

La plataforma ha sido mejorada siguiendo las necesidades y experiencias de los docentes, ofrece una interfaz intuitiva y clara de administración de espacios; su gestión o manejo no requiere ninguna técnica especial, esta se instala rápidamente y el uso de cualquier navegador web permite manejar las diferentes partes del curso y la admisión de los usuarios fluye con facilidad [6].

1.3.2. Moodle

Moodle es un Sistema de Gestión de Cursos aunque también es conocido por otros nombres, como LMS o Entorno de Aprendizaje Virtual. Esta herramienta permite a los profesores y educadores la creación de cursos en línea, aunque también puede ser utilizado como herramienta de trabajo colaborativa. El objetivo es que el usuario sólo necesite un navegador web en su ordenador y una conexión a Internet para interactuar con la herramienta [7].

El diseño y el desarrollo de Moodle está guiado por una filosofía particular de aprendizaje, una manera de pensar que recibe el nombre de “pedagogía social constructorista”. Esta filosofía está basada en 4 conceptos principales:

- **Constructivismo.** La teoría constructivista, atribuida al filósofo Jean Piaget, sostiene que las personas construyen nuevos conocimientos de manera activa al tiempo que interactúan con su entorno siguiendo un proceso de asimilación y acomodación. Una persona asimilará un concepto cuando las experiencias sean alineadas con respecto al conocimiento previo de la persona. Por otra parte, el proceso de acomodación, es el proceso en el cual la persona debe acomodar los conocimientos previos a los nuevos conocimientos que ha adquirido.
- **Constructorismo.** El constructorismo afirma que el aprendizaje es más efectivo cuando se construyen cosas. Por ejemplo, durante la lectura de este informe, el lector puede tomar notas, aun cuando no vaya a utilizarlas posteriormente, la construcción de estas notas permitirá una mejor asimilación de los conceptos con sus propios conocimientos.
- **Constructorismo Social.** Este concepto extiende las ideas anteriormente descritas a un grupo social. Los individuos de este grupo social construyen artefactos para los otros individuos del grupo, creando de manera colaborativa una pequeña cultura de artefactos compartidos con significados compartidos [7].

Conectado y Separado. Esta idea profundiza en las motivaciones de los individuos dentro de una discusión. Una persona aplica el comportamiento separado cuando intenta mantenerse “objetivo” y tiende a defender sus propias ideas utilizando la lógica y encontrando puntos débiles en las ideas del oponente. Una persona utiliza un comportamiento conectado cuando aplica aproximación más empática que acepta subjetivamente, intentado escuchar y realizar preguntas, en un esfuerzo de comprender el otro punto de vista. El comportamiento construido está basado en que una persona es susceptible a ambas aproximaciones descritas y es capaz de elegir cuál de ellas es la apropiada en la situación actual [7].

1.3.3. Plataforma ELEVEN

Esta es una plataforma educativa multieditorial, integral y universal que brinda la posibilidad a los centros educativos de acceso a todo tipo de contenidos digitales, herramientas de gestión didáctica y funcionalidades docentes centralizadas en una aplicación [25].

ELEVEN dispone de una interfaz muy intuitiva y fácil de utilizar para cualquier tipo de usuario. Los centros que lo van a utilizar no necesitan un servidor debido a que esta funciona a través de Internet; se adapta al perfil de cada usuario; gracias a las funcionalidades de este, profesor, centro y estudiante tienen a su disposición una gran gama de posibilidades para dar respuesta a cada necesidad de manera personalizada y disfrutar de una plataforma a su medida [25].

Todo ello convierte a Eleven en la plataforma de educación virtual más avanzada del mercado. Esta plataforma permite que tanto el centro como el profesor puedan crear por primera vez contenidos propios, ejercicios, guías didácticas y repositorios personalizados de archivos, e integrarlos en los contenidos y recursos que ofrecen las diferentes editoriales, independientemente de cuál sea su formato, tipo, tecnología o sistema [25].

1.4. Editores de Texto

Editores WYSIWYG (What You See Is What You Get)

WYSIWYG es el acrónimo de What You See Is What You Get ("lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso. En el caso de editores de HTML este concepto se aplica a los que permiten escribir la página sobre una vista preliminar similar a la de un procesador de textos, ocupándose en este caso el programa de generar el código fuente en HTML [22].

TinyMCE

TinyMCE es un editor WYSIWYG de HTML liberado como código abierto bajo licencia LGPL por Moxiecode Systems AB. Tiene la capacidad de convertir los campos TEXTAREA de HTML y otros elementos HTML en instancias del editor. TinyMCE es muy fácil de integrar en otros sistemas de gestión de contenidos [24].

Algunas características del editor TinyMCE son:

- ❖ Fácil de integrar.
- ❖ Los temas y plugins, bloque de elementos no válidos y los atributos de la fuerza.
- ❖ Soporte multilinguaje utilizando paquetes de idioma [24].

CKEditor

CKEditor es un editor de texto para ser utilizado dentro de las páginas web. Es un editor WYSIWYG, lo que significa que el texto que se está editando es lo que parece lo más similar posible a los resultados cuando los usuarios tienen que publicar. Reúne las características de edición web común que se encuentran en aplicaciones de edición de escritorio como Microsoft Word y OpenOffice [23].

CKEditor ofrece el soporte del navegador mejor compatibilidad. El editor se comporta de la misma manera en todos los navegadores, incluido Internet Explorer, Firefox, Safari, Chrome y Opera. Todos los navegadores gestionan y ofrecen el mismo contenido [23].

CKEditor hereda la calidad y los rasgos fuertes que fueron utilizados para encontrar en FCKeditor, en un producto mucho más moderno, agregó por decenas de nuevos beneficios, como la accesibilidad y el rendimiento final [23].

1.5. Metodologías para el desarrollo del software

1.5.1. Rational Unified Process (RUP)

El Proceso Unificado de Desarrollo (por sus siglas en inglés *RUP*, *Rational Unified Process*) está clasificado dentro de las metodologías de desarrollo tradicionales o pesadas, pero gracias a su flexibilidad puede ser adaptada para realizar una planeación y documentación mínima. Es, sin dudas, la metodología más utilizada para realizar el análisis, diseño, implementación y documentación de sistemas basados en programación orientada a objetos [20].

RUP divide en 4 fases el desarrollo del software, estas son: inicio, la que define el modelo del negocio, el alcance y los límites del proyecto; elaboración, en la cual se define, valida y cimienta la arquitectura; construcción, donde se desarrolla el producto y transición, la encargada de poner el producto en manos de los usuarios [20].

Por cada una de estas fases se transita a través de nueve flujos de trabajo, para los que RUP propone una serie de artefactos, actividades y roles que varían de un flujo de trabajo a otro. Su ciclo de vida está caracterizado por ser:

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a

través de los requerimientos. A partir de aquí, los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. La arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.

Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque hace hincapié en algunos más que en otros de acuerdo con la fase [20].

Otras características que hacen que esta metodología sea la ideal para guiar el desarrollo del sistema son:

- Orientado a objetos.
- Utiliza UML (Lenguaje Unificado de Modelado) como lenguaje de representación visual.
- Apto para desarrollar grandes y complejos proyectos.
- No es esencial la presencia del cliente en el desarrollo del producto.

Asociado al desarrollo se genera una gran cantidad de documentación que pueden ayudar a entender y mantener el sistema [20].

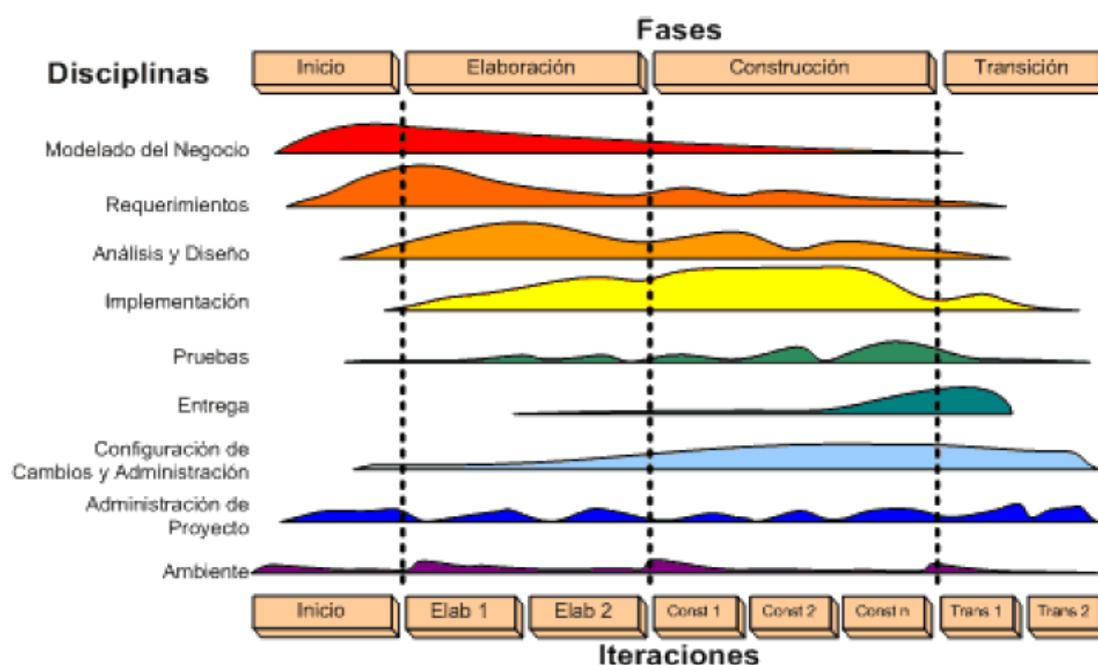


Figura 1.1 Fases y Flujos de Trabajo de RUP.

1.5.2. Extreme Programming (XP)

Esta es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado [9].

XP se caracteriza por ser una metodología ágil, la cual ofrece al cliente el software que necesita y cuando lo necesita además de combinarse en él las mejoras prácticas para el desarrollo de software. Es uno de los procesos ágiles más destacados en el desarrollo de software, diferenciándose de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Tiene como meta la disminución de costes y su objetivo principal es satisfacer a los clientes y potenciar al máximo el trabajo en grupo, además los jefes de proyecto, los desarrolladores y el cliente son parte del equipo de desarrollo. En el caso de los desarrolladores deben conocer que durante el ciclo de desarrollo de un proyecto de software los cambios aparecen, cambian los requisitos, las reglas del negocio, el personal y la tecnología, por lo tanto, este no es el principal problema ya que va a suceder, en donde reside el principal problema es en la capacidad que tengan para adaptarse a esos cambios [9].

En un proyecto desarrollado con esta metodología se definen 4 variables:

Coste: Máquinas, especialistas y oficinas.

Tiempo: Total y Entregas.

Calidad: Externa e Interna.

Alcance: Intervención del cliente.

Esta metodología trata de minimizar la complejidad de un proyecto y de enfocarse directamente hacia el objetivo, hace uso de las relaciones interpersonales y la rapidez de reacción. Se caracteriza por llevar a cabo pruebas unitarias, basadas en pruebas hechas a los procesos de mayor importancia. Una de las características más interesantes de esta metodología es que permite la programación en parejas, la cual consiste en que dos programadores trabajen en una misma estación, hace más eficiente el desempeño, ya que un programador se auxilia de su compañero [9].

El ciclo de vida ideal de XP consta de seis fases:

- Exploración
- Planificación de la entrega
- Iteraciones
- Producción
- Mantenimiento
- Muerte del proyecto

¿Qué metodología utilizar para solucionar el problema?

XP es una metodología que tiene como característica esencial potenciar al máximo el trabajo en grupo, en el que tanto los jefes del proyecto como los clientes y desarrolladores son parte del equipo de trabajo y se encuentran involucrados en el desarrollo del producto final.

En esta situación el cliente no forma parte del equipo de desarrollo por lo que la metodología no cumpliría con algo de extrema importancia para lograr un software con la calidad requerida.

RUP mantiene al equipo encauzado en la producción de software operativo a tiempo, con las características y calidad requerida, este además posee las mejores prácticas probadas en la industria, incorporando las enseñanzas de líderes así como la experiencia de una amplia gama de proyectos.

Se decide escoger la metodología RUP para el desarrollo del software por las características presentadas anteriormente, así como que basa su trabajo principalmente en la documentación del software, expone también un conjunto de actividades orientadas a visualizar, especificar, construir y documentar los artefactos necesarios para el desarrollo de software con la calidad que el cliente lo necesite; sumando a estas características también tenemos las políticas aplicadas por el proyecto que realiza el software que en la captura de requisitos y negocios de acuerdos con los clientes decidió que la metodología más factible a utilizar debería ser RUP.

1.6. Lenguaje Unificado de Modelado (UML) como propuesta de modelación a utilizar en la solución

Es un lenguaje gráfico para visualizar, especificar, construir y documentar artefactos de un sistema de software [20].



Figura 1.2 Lenguaje Unificado de Modelado.

UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables [20].

Características principales de UML.

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Mediante UML se pueden especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.)

- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.

UML proporciona a los desarrolladores un vocabulario que incluye tres categorías: elementos, relaciones y diagramas [20].

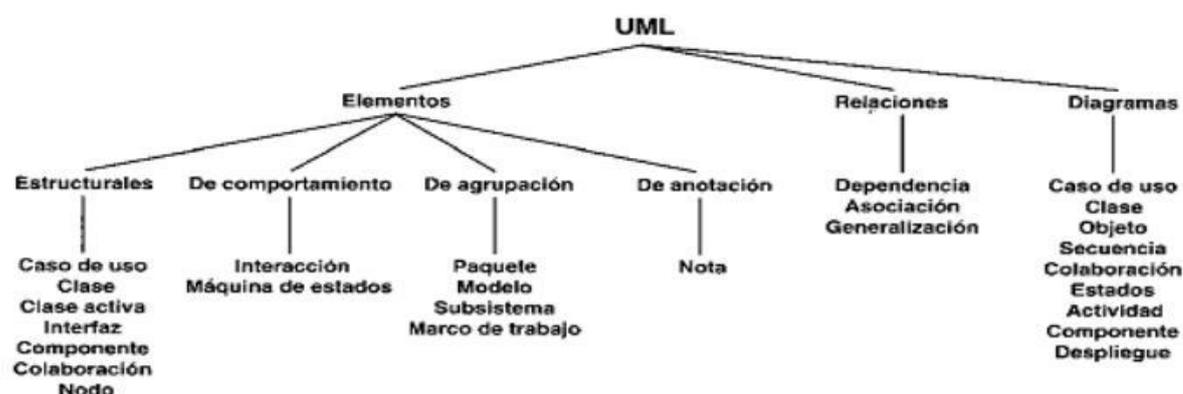


Figura 1.3 UML sus relaciones, elementos y diagramas.

1.7. Lenguajes y herramientas seleccionados para el desarrollo

Se hará uso de la tecnología web para solucionar el problema en cuestión debido a las ventajas que brinda en cuanto a eficiencia, con el propósito de que sea flexible y rápido; además de evitar instalaciones complicadas en los sistemas informáticos, la gestión de sistemas se simplifica mucho y el soporte al usuario puede darse a distancia.

Lenguajes de programación empleados

1.6.1. Del lado del cliente

XHTML

El lenguaje XHTML es muy similar al lenguaje HTML. El XHTML no es más que una adaptación de HTML al lenguaje XML. Técnicamente, HTML es descendiente directo del lenguaje SGML (Lenguaje de Marcado Generalizado), mientras que XHTML lo es de XML (que a su vez también es descendiente de SGML). Definiéndolo de forma sencilla, “HTML es lo que se utiliza para crear todas las páginas web de Internet”. Más concretamente, HTML es el lenguaje con el que se “escriben” la mayoría de las páginas web. El propio W3C define el Lenguaje de Marcado de Hipertexto como “un

lenguaje reconocido universalmente y que permite publicar información de forma global”. Desde su creación, el lenguaje HTML ha pasado a ser un lenguaje utilizado exclusivo para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica [10].

CSS

Lenguaje de hojas de estilo elaborado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS (hojas de estilo en cascada) es la vía más factible de separar los contenidos y su presentación; es indispensable para crear páginas web de alta complejidad. Separar la definición de los contenidos y la definición de su aspecto proporciona numerosas ventajas, debido a que este exige crear documentos HTML/XHTML bien definidos y con significado completo (conocidos también por “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes [11].

JavaScript

Es un lenguaje de programación que se utiliza esencialmente en la creación de páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Es decir, los programas escritos con JavaScript se pueden probar claramente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems [26].

1.6.2. Del lado del servidor

PHP

Es un lenguaje de script interpretados en el lado del servidor que se utiliza para la generación de páginas web dinámicas, embebido en páginas HTML y ejecutado en el servidor. Gran parte de la sintaxis del mismo ha sido tomada de lenguajes como C, Java, Perl con algunas características específicas propias del mismo. El lenguaje PHP tiene como meta permitir rápidamente al equipo de desarrollo la generación dinámica

de páginas; este no es un lenguaje de marcas como puede serlo HTML, XML o WML, al ser un lenguaje libre dispone de una gran variedad de características que lo convierten en la herramienta ideal para el desarrollo de páginas web dinámicas, las cuales se muestran a continuación [16]:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores que permite que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que se pueda realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas [16].

Framework

El término framework se traduce en una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Es propio de un framework incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para facilitarle el desarrollo y unir los diferentes componentes de un proyecto [17].

En otras palabras un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar el código ya existente y promover buenas prácticas de desarrollo con el uso de patrones. Un framework Web, por tanto, se

puede definir como un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web [17].

Por las características que se mencionarán a continuación se decidió escoger Symfony como framework para el desarrollo del software que necesita el cliente [18].

Symfony

Este framework se encarga de simplificar el desarrollo de una aplicación a través de la automatización de algunos de los patrones utilizados para resolver las tareas comunes, proporciona una estructura al código fuente, forzando a los desarrolladores a realizar un código más legible y más fácil de mantener [19].

Este framework es un facilitador de la programación de aplicaciones puesto que encapsula operaciones complejas en instrucciones sencillas. Está desarrollado completamente con PHP 5; se ha experimentado con él en números proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel, el mismo es compatible con la mayoría de los gestores de bases de datos, tales como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft [19].

Entre sus principales características están:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de las mejores prácticas y patrones de diseño para la web
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros [19].

Jquery

jQuery es una rápida y concisa colección de JavaScript que simplifica el documento HTML que atraviesa, el control de eventos, animación, y las interacciones Ajax para el desarrollo web rápido. jQuery está diseñado para cambiar la forma en que escribes JavaScript [21].

Las aplicaciones en internet son cada vez más complejas, ya que incorporan efectos visuales, drag and drop, auto-completar, animaciones etc. el desarrollar todos estos conceptos desde cero puede resultar complicado sobre todo si tenemos que presentar la solución con muy poco tiempo, en este tipo de situaciones el empleo de librerías como el jQuery facilitan el desarrollo de la aplicación. Otra ventaja paralela es despreocuparse cuando se codifica en la compatibilidad de navegadores, ya que la librería resolverá esto [21].

jQuery ahorra muchas líneas de código. Muestra transparente el soporte de la aplicación para los navegadores principales. Provee de un mecanismo para la captura de eventos. Brinda un conjunto de funciones para animar el contenido de la página en forma muy sencilla. Integra funcionalidades para trabajar con AJAX [21].

ORM

El ORM (Object Relation Mapper o Mapeo de Objetos Relacional) es una técnica de programación que brinda la posibilidad de convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, o sea, las tablas de la base de datos pasan a ser clases y los registros objetos que podemos manejar con facilidad.

Utilizar un ORM tiene una serie de ventajas que facilita tareas comunes y de mantenimiento tales como:

1. **Reutilización:** La principal ventaja que aporta un ORM es la reutilización permitiendo llamar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones.
2. **Encapsulación:** La capa ORM encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función.

3. **Portabilidad:** Utilizar una capa de abstracción permite cambiar en mitad de un proyecto de una base de datos MySQL a una Oracle sin ningún tipo de complicación. Esto es debido a que no se utiliza una sintaxis MySQL, Oracle o SQLite para acceder al modelo, sino una sintaxis propia del ORM utilizado que es capaz de traducir a diferentes tipos de bases de datos.
4. **Seguridad:** Los ORM suelen implementar mecanismos de seguridad que protegen nuestra aplicación de los ataques más comunes como SQL Injections.
5. **Mantenimiento del código:** Gracias a la correcta ordenación de la capa de datos, modificar y mantener nuestro código es una tarea sencilla.

Doctrine

Doctrine es una librería para PHP que permite trabajar con un esquema de base de datos como si fuese un conjunto de objetos, y no de tablas y registros; Está inspirado en Hibernate, que es uno de los ORM más populares y grandes que existen y brinda una capa de abstracción de la base de datos muy completa. La característica más importante es que brinda la posibilidad de escribir consultas de base de datos en un lenguaje propio llamado Doctrine Query Language (DQL).

Posee características como la generación automática del modelo. En presencia del trabajo con ORM se necesita crear un conjunto de clases que serán vinculadas al esquema de la base de datos de forma automática con un motor ORM.

Posibilita el trabajo con YAML, brinda la posibilidad de definir el mapeo de tablas y sus relaciones. Esto se puede hacer con código PHP o con YAML, que es un formato de serialización de datos legible por humanos muy usado para este fin.

Otra característica es las relaciones entre entidades ya que una vez que se ha definido el modelo (o se ha creado de forma automática) con las tablas y sus relaciones, resulta fácil acceder y moverse por entidades relacionadas.

Propel

Propel es un ORM para PHP que facilita la labor de desarrollo de aplicaciones web, gracias a la capa que transforma el tratamiento de la BD mediante objetos, con la que se puede recuperar, insertar y modificar datos. La preocupación por las conexiones de la base de datos y escribir SQL o escapar los datos o realizar casting, están de más;

solamente es necesario definir la base de datos en formato XML u obtener una definición desde una base de datos existente.

Ambos ORMs tienen bastantes características básicas similares, ya que soportan cualquier operación usual en un CRUD (Create, Retrieve, Update and Delete), ya sea desde crear un nuevo registro o actualizar los registros existentes. Además, los dos pueden generar las clases PHP del modelo, Propel basado en XML y Doctrine basado en YAML, y ambos soportan la generación de sus respectivos markups desde una base de datos existente. Pero Doctrine actualmente ha dejado atrás a Propel ya que soporta Data fixtures y migraciones, caching, events, pagination y command line interface, con lo se podría decir que en estas características avanzadas Doctrine supera a Propel. Además, las versiones actuales de symfony utilizan como ORM principal Doctrine, por lo que se define el uso del ORM doctrine en la aplicación a desarrollar [20].

1.8. Sistema Gestor de Base de Datos

1.7.1. PostgreSQL

Poderoso sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de confiabilidad, integridad de datos y corrección. Funciona en los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows [14].

Este sistema tiene soporte completo claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios idiomas). Es compatible con el almacenamiento de objetos binarios, incluyendo a esto imágenes, sonidos o vídeos, la documentación es de carácter excepcional, altamente escalable, tanto en la enorme cantidad de datos que puede manejar como en la cifra de usuarios concurrentes que puede permitir [14].

Posee características que propician que sea este el sistema de base de datos escogido para la realización del proyecto, estas son:

- El tamaño máximo de la base de datos es ilimitado.
- El tamaño máximo de una tabla es de 32 Terabytes.
- El tamaño máximo de un campo es de 1.6 Terabytes.
- El máximo de filas por tabla es ilimitado.

- El máximo de columnas por tabla es 250 – 1600 dependiendo de los tipos de columna.
- Los índices máximos por el cuadro son ilimitados [14].

1.9. Entorno de desarrollo integrado seleccionado

1.8.1. NetBeans IDE

Esta herramienta es para programadores con el objetivo de escribir, compilar y depurar programas, es libre y gratuito sin restricciones en su uso [15].

El NetBeans 6.8 brinda soporte para PHP 5.3, también para Symfony. Algunas de las posibilidades que hasta ahora brinda el IDE se muestran a continuación:

Para PHP 5.3:

- Soporte para las palabras clave (keywords).
- Soporte para namespaces.
- Soporte para closures.

Para Symfony:

- Ejecución de comando Symfony.
- Limpiar la caché.
- Ayuda para encontrar comandos y para conocer los parámetros.
- Muestra toda la ejecución de cada comando (creación de ficheros, directorios, etc.)
- Configuración de atajos del teclado [15].

1.10. Herramienta CASE para el modelado UML

1.9.1. Visual Paradigm

Herramienta que soporta todos los diagramas UML y diagrama entidad relación, esta para UML brinda amplias características de modelado de casos de uso incluyendo la función completa de UML, diagramas de casos de uso, flujos de evento editor, casos de uso [13].

Esta herramienta genera una documentación del sistema con un diseñador de plantilla, los analistas pueden estimar las consecuencias de los cambios en los diagramas de análisis tales como la matriz, además de todo lo antes planteado esta herramienta permite la generación de código Java [13].

1.11. Herramientas para el Modelado de Prototipos

1.10.1. Balsamiq Mockups v1.6.58

Balsamiq Mockups es una aplicación de escritorio, posee una interfaz fácil de usar con una colección de botones bastante grande y puedes crear casi cualquier programa, es una gran herramienta para generar de una forma ágil y sencilla wireframes. Presenta prototipos de diseños para web, móvil, de una forma realmente original, como un paso previo a empezar a montar los diseños en Photoshop o Gimp.

Balsamiq dispone de más de 75 modelos ya definidos con los diferentes elementos de las interfaces de usuario para montar el prototipo, y también permite añadir por los usuarios prototipos de otras librerías de elementos.

1.10.2. Pencil Project

Pencil Project es un proyecto libre para la creación de diagramas y prototipos GUI de forma rápida y sencilla, está bajo la licencia GPL v2, es multiplataforma. Una de sus características más importantes es la capacidad de alinear objetos para que el conjunto que se realiza quede ordenado.

Los proyectos creados en esta herramienta se guardan en XML, siendo capaz también de exportar nuestros diseños a diferentes formatos tales como PNG, PDF, ODT o HTML [18].

1.10.3. Dia

Esta herramienta es un editor de digramas que posee funcionalidades para crear los mismos o que puedan ser modificados, sin conocimientos previos de la misma. Dia contiene herramientas para el dibujo que permiten introducir elementos geométricos a nuestras composiciones, posibilitando la edición de sus propiedades, brinda la posibilidad de organizar los diagramas y los sistemas de capas.

Permite exportar los diseños realizados a los formatos más conocidos, todo lo antes mencionado demuestra que la herramienta es fácil de usar, que tiene una gran variedad de herramientas de dibujo; aunque su interfaz podría mejorarse.

CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

En el desarrollo de un sistema computarizado se tiene la necesidad de esclarecer todos los conceptos implicados en el tema y las relaciones de ellos utilizando un lenguaje de fácil entendimiento para los implicados con el mismo; todo esto conlleva a que en este capítulo se realice la descripción de la propuesta de solución garantizando una mejor comprensión. En el documento queda plasmado el Modelo de Dominio que agrupa los conceptos fundamentales con los que se trabaja en la solución propuesta. En el capítulo se identifican los requisitos funcionales y no funcionales y los casos de uso significativos que regirán el desarrollo del sistema.

2.2. Modelo de Dominio

2.2.1. Análisis de los conceptos del Dominio

- ✓ **Editor Digital:** Es la persona facultada para trabajar en el área de la gestión de las páginas.
- ✓ **Página:** Conjunto de contenidos agrupados de acuerdo a una materia dada.
- ✓ **Plantilla:** Es un elemento que puede ser creado por el usuario a su gusto para estructurar el contenido dentro de la página como tal
- ✓ **Palabra o frase clave:** Elemento que estará insertado en cualquiera de los textos pertenecientes a capítulos, temas, subtemas o páginas que contienen cada uno de estos y estará relacionado con alguno de los elementos anteriormente mencionados.
- ✓ **Recurso:** Elemento que sirve de apoyo al contenido que se encuentra dentro de la página.
- ✓ **Recurso multimedia:** Se entiende por recursos multimedia la información presentada a partir de cualquiera de los siguientes 5 formatos digitales o sus combinaciones: Texto o hipertexto, Video digital, Sonido digital, Animaciones e Imagen fija.
- ✓ **Imagen:** Elemento de ilustración gráfica.
- ✓ **Sonido:** Archivos de sonido, accesibles a través del servidor streaming.
- ✓ **Diaporama:** Imágenes de video o fijas con audio.
- ✓ **Video:** Imágenes de video o fijas con audio, publicadas a través de la tecnología streaming.
- ✓ **Animaciones:** Archivo digital con formato *.flv.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

- ✓ **Recurso estructural:** Se trata de un conjunto de recursos cuya función es la de auspiciar un andamiaje de apoyos didáctico que posibilite establecer elementos del contenido con marcada intencionalidad pedagógica.
- ✓ **Aplicación práctica:** Elemento de interés para el usuario, que apoyan el proceso de enseñanza-aprendizaje y aumentan la motivación del usuario al mostrarle como se aplica en la práctica el contenido aprendido; se presenta como una descripción acompañada, como recursos multimedia o una unión de las mismas.
- ✓ **Investiga y aprende:** Sección en la cual el usuario puede encontrar temas que serán objeto de investigaciones.
- ✓ **Sabías que:** Se empleará este recurso para enfatizar o llamar la atención sobre contenidos tendientes a provocar curiosidad, situaciones problemáticas y que están llamados a trabajar sobre la esfera motivacional más que sobre la esfera cognitiva.
- ✓ **Recuerda que:** Elemento que comprende la unión de varios elementos interactivos, es un resumen para que el estudiante recuerde algunos casos específicos.
- ✓ **Saber más:** Resumen conformado por texto y elementos multimediales para recordar un tema.
- ✓ **Historia:** Personalidades históricas y hechos históricos relacionados con el contenido que se trata.
- ✓ **Recurso interactivo:** Sustentado en el modelo de comunicación Shannon Weber, por interactividad en el marco informático, se tratará de entender tanto la interactividad hombre-máquina como a que se produce entre hombres vehiculada por la máquina (hombre-hombre mediada por la máquina).
- ✓ **Preguntas interactivas:** Se trata del planteamiento de interrogantes derivadas de un conjunto predeterminado de variantes que serán evaluadas por la computadora.
- ✓ **Simuladores:** Son recursos altamente interactivos que modelan fenómenos, leyes o procesos en correspondencia con la variación de parámetros de entrada por parte del usuario.
- ✓ **Secuencias interactivas:** Se trata de propuestas de flujos de información caracterizadas por la aparición paulatina de información de carácter multimedia y la toma de decisión por parte del usuario como condición para dar continuidad al discurso.
- ✓ **Enfrenta retos:** Sección que agrupa varias sub-secciones, con el objetivo de agrupar diferentes tipos de ejercicios.

2.2.2. Diagrama del Modelo de Dominio

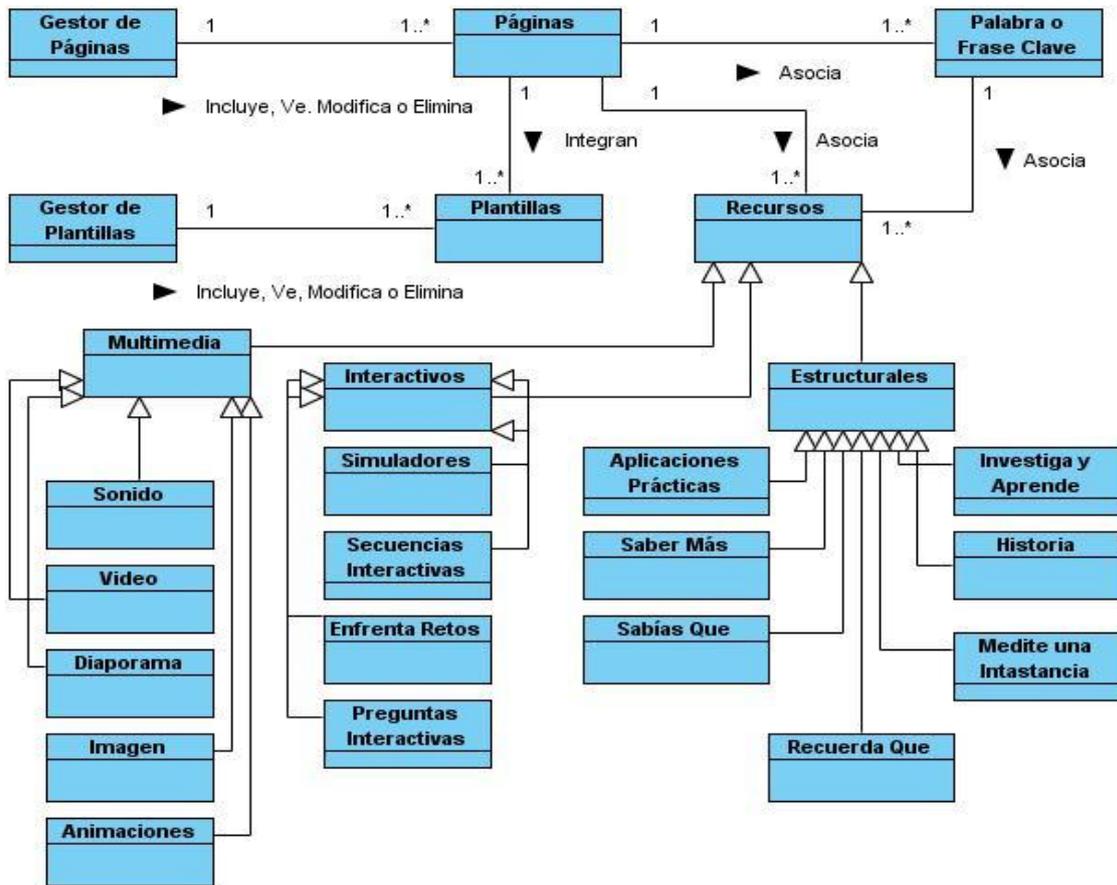


Figura 2.1 Modelo de dominio.

2.2.3. Descripción del Sistema Propuesto

El componente propuesto tiene como objetivo principal la creación de páginas que almacenen los contenidos de una materia dada, permitiendo que el estudiante pueda obtener el contenido para su estudio individual. El componente contará con plantillas predeterminadas que facilitarán la gestión de las páginas permitiendo al editor o cualquier usuario que esté creando la página la facilidad a la hora de asociar un recurso a la página para lograr que el contenido llegue a su fin con un ambiente interactivo y ameno para el usuario. El componente para la gestión de las páginas tendrá un editor de texto que le permitirá al usuario hacer una palabra o frase clave dentro del texto que esté editando en ese momento, brindando la posibilidad de asociarle recursos a esta palabra tales como animaciones, sonidos, diaporamas, imágenes, aplicaciones prácticas, entre otros.

2.3. Requerimientos del software

2.3.1. Requerimientos Funcionales

Los requerimientos funcionales definen las capacidades o condiciones que el sistema debe ser capaz de cumplir.

RF 1: Gestionar Página.

RF 1.1 Incluir página: Se selecciona la opción de incluir una nueva página, para ello una vez que se encuentre en el escritorio de trabajo para realizar esto se selecciona una plantilla predeterminada logrando con esto una mejor estructuración del contenido dentro de la página.

RF 1.2 Vista previa de la página: Para ello debe existir al menos una página creada. Se da un clic encima de la página que desea obtener su vista previa y se mostrará la misma.

RF 1.3 Modificar página: Para ello debe existir al menos una página creada. Se da clic en la página que desea modificar y la misma se muestra con los campos editables de las plantillas. Se selecciona la opción actualizar para modificar los datos de la página.

RF 1.4 Eliminar página: Para ello debe existir al menos una página creada. Se da clic derecho encima de la página que se quiere eliminar y la misma se oculta.

RF 2: Gestionar Palabra o Frase Clave.

RF 2.1 Incluir palabra o frase clave: Al seleccionar esta opción se permite asociar recursos o un significado de palabras del glosario a una palabra o frase del texto para enriquecer la misma.

RF 2.2 Modificar datos de la palabra o frase clave: Para ello debe existir al menos una palabra o frase creada. Se selecciona se selecciona la palabra o frase y se da clic en el botón para asociar o desasociar recursos a la misma, actualizando los datos de la misma.

RF 2.3 Eliminar palabra o frase clave: Para ello debe existir al menos una palabra o frase creada. Se selecciona la palabra y con ello se selecciona el botón deshacer palabra o frase clave y esta deja de serlo.

RF 3: Gestionar plantillas.

RF 3.1 Incluir plantilla: Cuando se hace uso de esta acción se crea una nueva plantilla predeterminada para ser incluida dentro de la página y facilitar la estructuración del contenido de la página.

RF 3.2 Ver plantilla: Para ello debe existir al menos una plantilla creada. Se selecciona la plantilla y se muestran los datos de la misma.

RF 3.3 Modificar plantilla: Para ello debe existir al menos una plantilla creada. Se selecciona la plantilla y se modifican los datos de la misma actualizando los cambios en esta.

RF 3.4 Eliminar plantilla: ello debe existir al menos una plantilla creada. Se selecciona la plantilla y se elimina la misma del listado de plantillas.

RF 4: Editar Texto: Cuando se selecciona esta opción se edita el texto contenido en las plantillas predeterminadas y a su vez los recursos asociados a estas.

RF 5. Editar Texto Acotado: Esta opción se selecciona en el caso de que se quiera modificar algún texto en el área de recursos o en el área del docente.

RF 6. Listar Páginas: Esta opción se puede observar a la izquierda del escritorio de trabajo donde se encuentran las páginas que están creadas en el índice.

RF 7. Listar Plantillas: Esta opción se puede observar a la derecha del escritorio de trabajo donde se encuentran las plantillas que están creadas para ayudar con la estructuración del contenido de las páginas.

RF 8. Mostrar Palabra o Frase Clave: Para ello debe existir al menos una palabra o frase creada. Si la palabra tiene asociado recursos se levanta un poput encima del escritorio de trabajo que muestra los que están asociados a la misma.

2.3.2. Requerimientos No Funcionales

Los requerimientos no funcionales brindan la posibilidad de definir propiedades con las que el producto debe contar, lo cual propicia que este sea interesante, rápido y confiable.

Luego de conocer lo que funcionalmente el componente debe ser capaz de hacer entonces se hace posible definir como debe comportarse, es decir, las cualidades que

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

debe tener el mismo; este tipo de propiedades son de gran importancia para los usuarios a la hora de valorar la aceptación del producto respecto a su seguridad, usabilidad y cumplimiento de las funcionalidades definidas.

RNF 1: Seguridad y confiabilidad.

- ✓ El componente brindará dependiendo del perfil de usuario que se ingrese a través de usuario y contraseña el acceso a las funcionalidades del mismo.
- ✓ Se utilizarán ofuscadores de código fuente para garantizar la seguridad de este.
- ✓ Se asegurará la plataforma de los ataques de hackers tales como SQL Injection, mediante la escritura de código seguro.

RNF 2: Restricciones de diseño

- ✓ PHP 5.3.3 como lenguaje de programación.
- ✓ Symfony 1.4 como marco de trabajo base de desarrollo.
- ✓ Netbeans 6.9 como IDE.
- ✓ Apache 2.2.16 como servidor web.
- ✓ PostgreSQL 8.4 como sistema gestos de base de datos.

RNF 3: Interfaz, usabilidad.

- ✓ Interfaz amigable sugerente al usuario.
- ✓ El componente brindará una clara y correcta organización de la información permitiendo la interpretación correcta e inequívoca de la misma.

RNF 4: Interfaces de Hardware

- ✓ Las computadoras que brindarán servicio cliente del sistema deben tener las prestaciones brindadas por una Pentium 4, contando al menos con 512 MB de RAM y 200 MB de espacio en disco.

RNF 5: Legales, Derecho de autor.

- ✓ Una vez terminado el componente debe ser sometido a una evaluación y certificación por parte del cliente del producto.

2.4. Modelo de Casos de Uso del Sistema

2.4.1. Descripción de los actores del sistema

Actor	Descripción
Editor Digital	Es la persona facultada para la gestión de las páginas la misma después de crear las plantillas necesarias para la estructuración del contenido puede crear una página la cual puede incluir dentro de ella palabras o frases clave que pueden tener recursos asociados al contenido de la página; además de ello este actor es que edita el texto dentro de la plantilla que seleccione para la confección de la página. Un administrador central puede comportarse como un Editor Digital.
Editor de Texto	Es la persona que tiene permisos para utilizar el editor de texto acotado a la hora de gestionar un recurso o algún material necesario dentro del módulo docente, entre otros. Puede comportarse como editor de texto el administrador central, el administrador local, el tutor, el estudiante, el director y el docente en dependencia de las acciones que este desee realizar y con qué privilegios se encuentre trabajando.

Tabla 2.1 Descripción de los actores del sistema.

2.4.2. Diagrama de actores del sistema

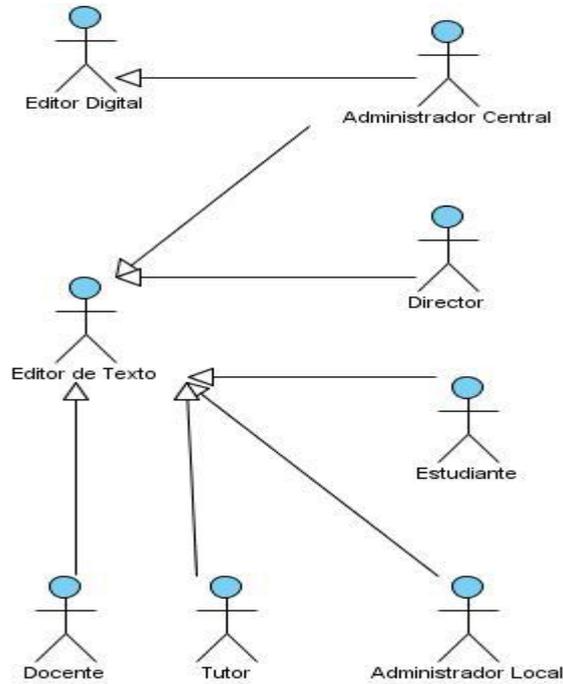


Figura 2.2 Diagrama de personas que intervienen en el sistema propuesto.

2.4.3. Diagrama de casos de uso del sistema

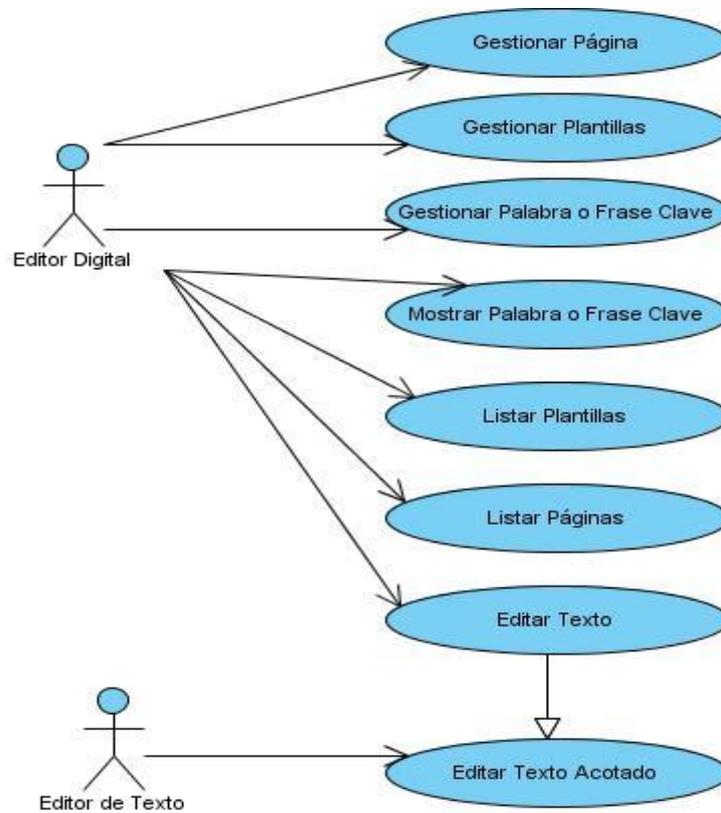


Figura 2.3 Diagrama de Casos de Uso del Sistema

2.4.4. Descripción de los casos de uso del sistema

Descripción del CU Gestionar Página.

Flujo básico

Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de realizar una acción sobre la página.	2. Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> • Incluir una nueva página. • Vista previa de la página. Ver Sección 1: "Vista previa de la página". • Modificar el contenido de la página. Ver Sección 2: "Modificar contenido de la página". • Eliminar la página. Ver Sección 3: "Eliminar página".
3. Selecciona la opción de incluir una nueva página.	4. Brinda la posibilidad de seleccionar una de las plantillas creadas por el usuario.
5. Selecciona la plantilla.	6. Permite editar el texto dentro de la plantilla seleccionada. Y brinda la posibilidad de asociar un recurso a la plantilla, dependiendo del tipo de esta.
7. Edita el texto de la plantilla seleccionada.	
8. Selecciona la opción de asociar un recurso a la plantilla seleccionada.	9. Muestra todos los recursos del tipo seleccionado asociados al índice de la materia. Y brinda la posibilidad de seleccionar el recurso y editar la vista previa del mismo de acuerdo a: <ul style="list-style-type: none"> • Ancho. • Alto. • Borde. • Color de borde. Y permite además: <ul style="list-style-type: none"> • Cambiar el recurso asociado a la plantilla. • Incluir una palabra o frase como caliente o clave en la plantilla. • Seleccionar una nueva plantilla. • Selecciona editar el texto dentro de una plantilla. • Guardar la página. • Guardar la página sin publicar. • Guardar la página y continuar, en cualquier momento. • Cancelar la operación en cualquier momento.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

10. Selecciona la opción de guardar la página.

11. Valida los datos.

12. Crea la página y publica su contenido.

13. Muestra un mensaje de información.

14. El caso de uso termina.

Flujo alterno

*.a El actor selecciona la opción de Cancelar.

Acciones del actor

Respuesta del sistema

*.a.1 Elimina los datos creados.

*.a.2 Muestra un mensaje de información.

*.a.3 Regresa a la vista anterior.

*.a.4 El caso de uso termina.

*.b El actor selecciona la opción de Guardar y continuar.

Acciones del actor

Respuesta del sistema

*.b.1 Valida los datos.

*.b.2 *Guarda los datos de la página y se mantiene en la misma para seguir editándola.*

*.b.3 Regresa al paso 4 del flujo básico.

*.b.4 El caso de uso termina.

*.c El actor selecciona la opción de Guardar sin publicar.

Acciones del actor

Respuesta del sistema

*. c.1 Valida los datos.

*. c.2 *Crea la página pero no publica su contenido.*

*. c.3 Muestra un mensaje de información.

*. c.4 Permite incluir una nueva página.

*. c.5 El caso de uso termina.

9.a El actor selecciona la opción de seleccionar una nueva plantilla.

Acciones del actor

Respuesta del sistema

9. a.1 Regresa al paso 4 del flujo básico.

9. a.2 El caso de uso termina.

9.b El actor selecciona la opción de incluir palabra o frase como caliente o clave.

Acciones del actor

Respuesta del sistema

9. b.1 Muestra *Ver CU Gestionar Palabra Caliente o Frase.*

9. b.2 Regresa al paso 9 del flujo básico.

9. b.3 El caso de uso termina

9.c El actor selecciona la opción de editar texto dentro de una plantilla.

Acciones del actor

Respuesta del sistema

9. c .1 Muestra *Ver CU Editar Texto.*

9. c.2 Regresa al paso 9 del flujo básico.

9. c.3 El caso de uso termina

Sección 1: “Vista previa de la página”

Flujo básico

Acciones del actor	Respuesta del sistema
1. Selecciona la opción de ver una vista previa de la página.	2. <i>Muestra la</i> página según los estilos del hiperentorno a la cual está asociada. Y permite: <ul style="list-style-type: none"> • Salir de la vista actual.
3. Selecciona la opción de salir de la vista actual.	4. Muestra la vista anterior. 5. El caso de uso termina.

Sección 2: “Modificar contenido de la página”

Flujo básico

Acciones del actor	Respuesta del sistema
1. Selecciona la opción de modificar los datos de la entidad.	2. Muestra el contenido de la página y brinda la posibilidad de cambiarlo. Permite además: <ul style="list-style-type: none"> • Actualizar los cambios. • Actualizar sin publicar los cambios. • Cancelar la operación en cualquier momento.
3. Modifica el contenido y selecciona la opción de actualizar los cambios.	4. Valida los datos. 5. <i>Actualiza y publica el contenido de la página.</i> 6. Muestra un mensaje de información. 7. Muestra la vista previa de la página .Ver Sección 2: “Vista previa de la página.” 8. El caso de uso termina.

Flujo alternativo

3.a El actor selecciona la opción de Cancelar

Acciones del actor	Respuesta del sistema
	3.a.1 Muestra un mensaje de información.
	3.a.2 El caso de uso termina.

Sección 3: “Eliminar página”

Flujo básico

Acciones del actor	Respuesta del sistema
1. Selecciona la opción de eliminar una	

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

página.	
	2. Muestra el mensaje de confirmación.
3. Selecciona la opción de aceptar.	
	4. <i>Oculto la página.</i>
	5. Muestra el mensaje de información.
	6. Regresa a la vista anterior actualizando los datos.
	7. El caso de uso termina.

Flujo alternativo

3.a El actor selecciona la opción de Cancelar	
Acciones del actor	Respuesta del sistema
	3.a.1 Regresa a la vista anterior.
	3.a.2 El caso de uso termina.

Las descripciones restantes se encuentran en **Anexo # 1: Descripción de los Casos de Uso del sistema.**

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. Introducción

En el presente capítulo se realiza el análisis y diseño que sustenta la propuesta de solución del componente a desarrollar, se realizan para ello los diagramas de clases del análisis y del diseño de los casos de uso identificados así como los diagramas de interacción correspondientes a cada uno de ellos.

Se definen los patrones de diseño y el patrón arquitectónico los cuales regirán el proceso de desarrollo. Con el fin de garantizar el almacenamiento de la información se modela la estructura de la base de datos correspondiente. El módulo iPages cuenta con seis casos de uso, de los cuales se trabajará en el capítulo con los más significativos de ellos.

En el proceso de análisis se tienen en cuenta los requisitos descritos en la captura de requerimientos, refinando y estructurando los mismos para obtener con ello una comprensión más precisa. Aún cuando en el modelo del análisis se pulen los requisitos no se tiene en cuenta el lenguaje de programación a utilizar en la construcción del sistema, la plataforma en la que se correrá la aplicación entre otras características que afectan al sistema, debido a que el objetivo principal del análisis es comprender cabalmente los requisitos del software a desarrollar y no precisar como se implementará el mismo, a diferencia del diseño que si pone énfasis en estos elementos.

El diseño ya pasa a convertirse en el centro de atención al final de la fase de elaboración y al comienzo de las iteraciones de la construcción. En el diseño se modela el sistema y se define la arquitectura que da soporte a todos los requisitos teniendo en cuenta los requisitos no funcionales y las restricciones que se le suponen.

3.2. Modelo del análisis

3.2.1 Diagrama de clases del análisis

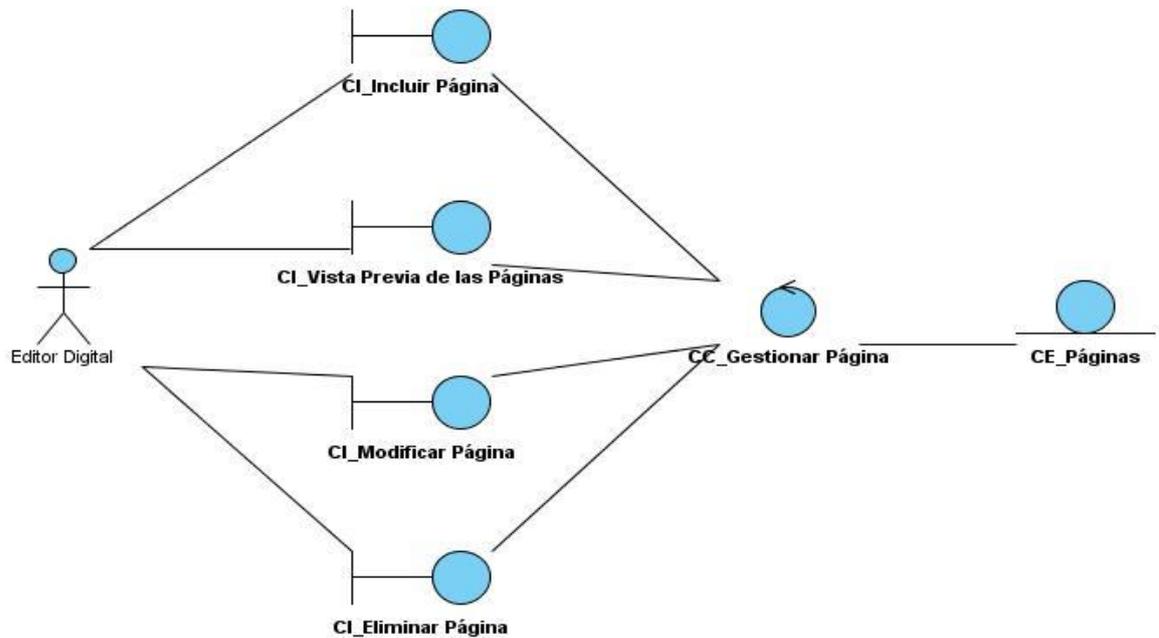


Figura 3.1 Diagrama de clases del análisis del CU Gestionar Página.

Los restantes diagramas de clases del análisis de los demás casos de uso se encuentran en **Anexo # 2: Diagramas de Clases del Análisis.**

3.2.2 Diagramas de colaboración del análisis



Figura 3.2 DC CU Gestionar Página Sección Incluir.

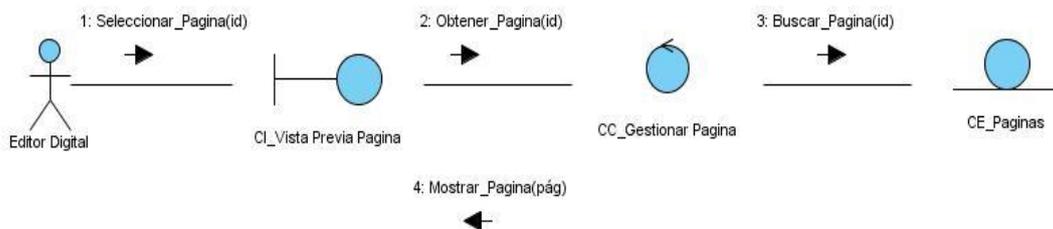


Figura 3.3 DC CU Gestionar Página Sección Vista Previa.

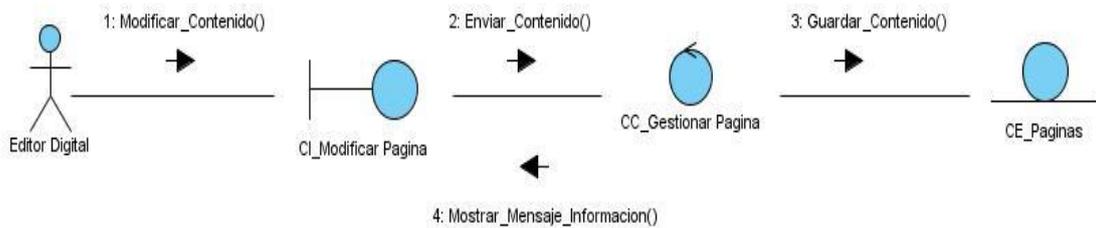


Figura 3.4 DC CU Gestionar Página Sección Modificar.

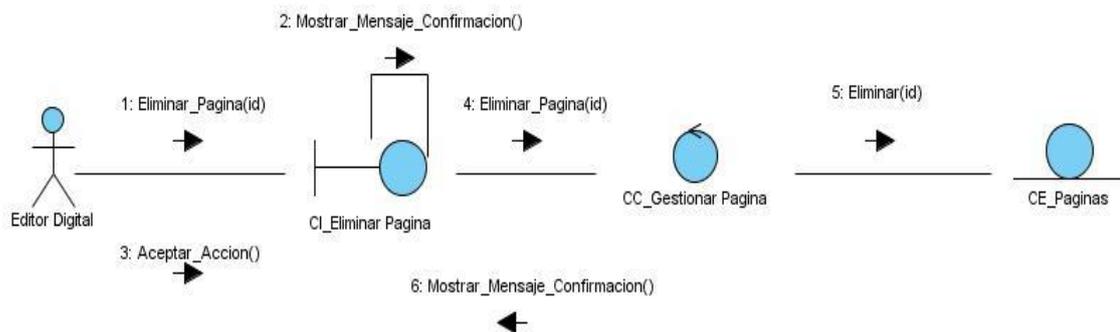


Figura 3.5 DC CU Gestionar Página Sección Eliminar.

Los restantes diagramas de colaboración del análisis de los demás casos de uso se encuentran en **Anexo # 3: Diagramas de Colaboración del Análisis.**

3.3. Patrón arquitectónico Modelo – Vista – Controlador en symfony

Symfony como framework está basado en el clásico patrón de diseño web conocido como arquitectura modelo-vista-controlador (conocido como MVC) el mismo está formado por tres niveles que se explican a continuación:

- ✓ **Modelo:** esta capa es la encargada de administrar todo lo relacionado con los datos del sistema, brinda respuesta a las peticiones de información sobre el estado de la aplicación (usualmente formulados desde la vista) y responde a instrucciones de cambiar de estado (habitualmente desde el controlador). Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- ✓ **Vista:** aquí se maneja la visualización de la información, es decir, gestiona lo relacionado con mostrar la información al usuario.

CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA

- ✓ **Controlador:** en esta capa se interpretan los eventos lanzados por la entrada del usuario, informando de los mismos al modelo y vista para que los cambios sean ejecutados correctamente.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

3.4. Modelo de diseño

3.4.1. Diagrama de clases del diseño

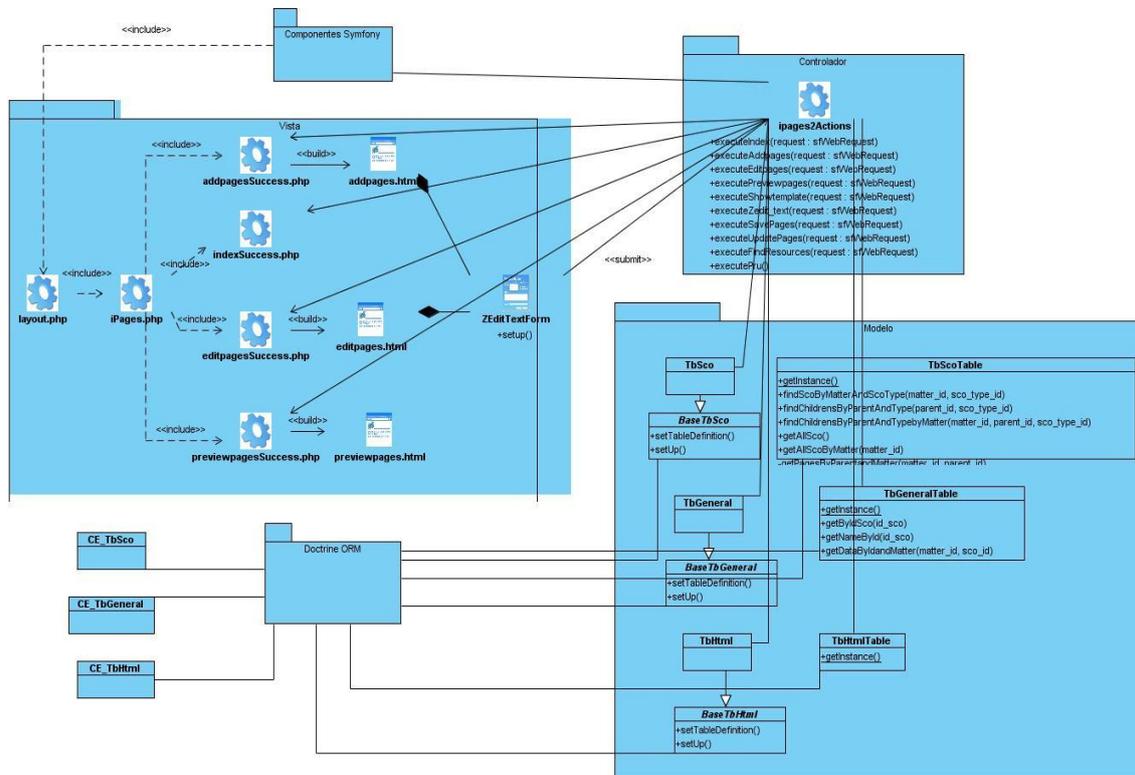


Figura 3.6 Diagrama de Clases del Diseño del CU Gestionar Página.

Los restantes diagramas de clases del diseño de los demás casos de uso se encuentran en **Anexo # 4: Diagramas de Clases del Diseño.**

3.5. Diseño de la base de datos

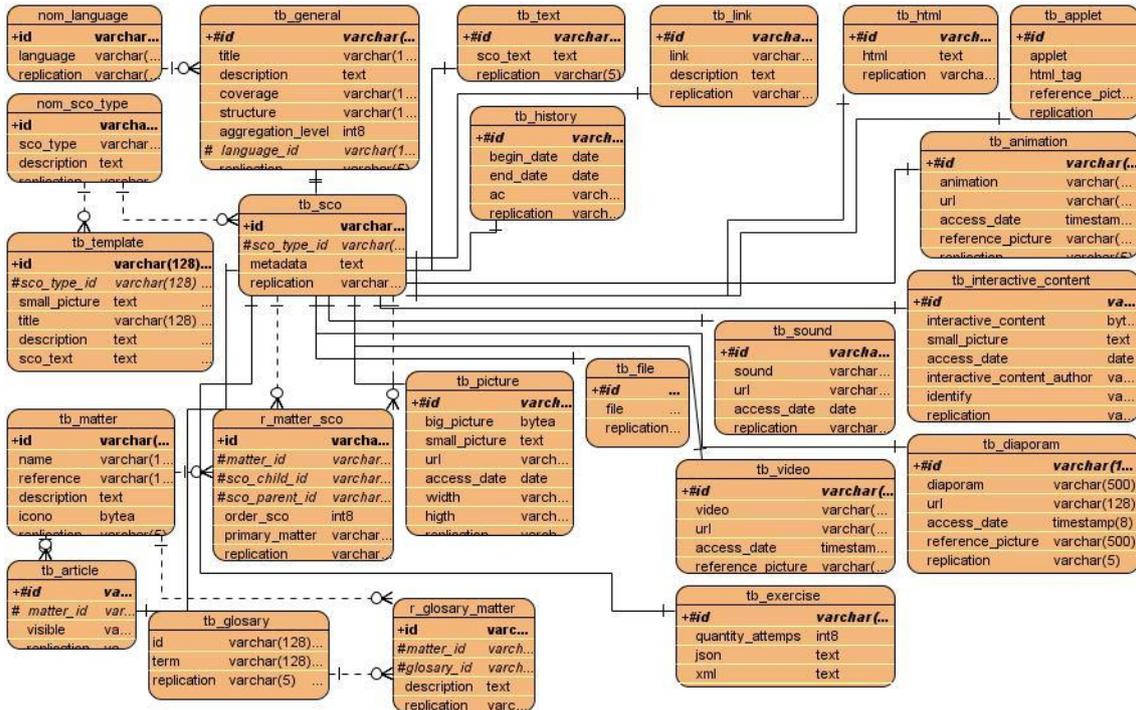


Figura 3.7 Diseño de la base de datos.

3.4.1. Descripción de las tablas

nom_sco_type		
Descripción: En esta tabla se guardan los distintos tipos de sco que se manejan, ya sean temas, subtemas, capítulos, entre otros.		
Atributo	Tipo	Descripción
id	VARCHAR(128)	
name	VARCHAR(128)	Atributo que permite guardar el nombre del tipo de sco.
sco_type	VARCHAR(128)	Atributo que permite guardar el identificador del tipo de sco.
description	VARCHAR(500)	Atributo que permite guardar una breve descripción del tipo de sco.
replication	varying(5)	Atributo que permite guardar los bits de replicación.
created_at	timestamp	Atributo que permite guardar la fecha de creada del nomenclador.
updated_at	timestamp	Atributo que permite guardar la fecha de la última modificación realizada al nomenclador.
deleted_at	timestamp	Atributo que permite guardar la fecha de eliminación del nomenclador.

Tabla 3.1 Descripción de la tabla nom_sco_type.

r_matter_sco
Descripción: Esta tabla es una relación entre las diferentes materias y los distintos sco donde se asocia por ejemplo un capítulo a una materia determinada, entre otras asociaciones que pueden existir.

CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA

Atributo	Tipo	Descripción
id	VARCHAR(128)	
matter_id(FK)	VARCHAR(128)	Atributo que permite guardar el id de la materia.
sco_child_id(FK)	VARCHAR(128)	Atributo que permite guardar el id del sco hijo.
sco_parent_id(FK)	VARCHAR(128)	Atributo que permite guardar el id del sco padre.
order_sco	NUMERIC(9, 0)	Atributo que permite guardar el orden de secuenciación en los scos.
replication	varying(5)	Atributo que permite guardar los bits de replicación.
created_at	timestamp	Atributo que permite guardar la fecha de creada de la relación.
updated_at	timestamp	Atributo que permite guardar la fecha de la última modificación realizada a la relación.
deleted_at	timestamp	Atributo que permite guardar la fecha de eliminación de la relación.

Tabla 3.2 Descripción de la tabla r_matter_sco.

tb_html		
Descripción: Esta tabla es una especialización sco donde se guarda el texto plano, es decir, el texto html como tal.		
Atributo	Tipo	Descripción
id	VARCHAR(128)	
html	TEXT	Atributo que permite guardar el contenido html de la página.
replication	varying(5)	Atributo que permite guardar los bits de replicación.
created_at	timestamp	Atributo que permite guardar la fecha de creación de la página.
updated_at	timestamp	Atributo que permite guardar la fecha de la última modificación realizada a la página.
deleted_at	timestamp	Atributo que permite guardar la fecha de eliminación de la página.

Tabla 3.3 Descripción de la tabla tb_html.

tb_sco		
Descripción: En esta tabla se almacenan todos los objetos de aprendizaje existentes.		
Atributo	Tipo	Descripción
id	VARCHAR(128)	
sco_type_id	varying(128)	Atributo que permite guardar el identificador del tipo de sco.
metadata	TEXT	Atributo que permite guardar los metadatos del sco en un XML.
replication	varying(5)	Atributo que permite guardar los bits de replicación.
created_at	timestamp	Atributo que permite guardar la fecha de creación del sco.
updated_at	timestamp	Atributo que permite guardar la fecha de la

CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA

		última modificación realizada al sco.
deleted_at	timestamp	Atributo que permite guardar la fecha de eliminación del sco.

Tabla 3.4 Descripción de la tabla tb_sco.

tb_general		
Descripción: Es una de las tablas LOM donde se guardan los datos generales de todos los objetos de aprendizaje existentes.		
Atributo	Tipo	Descripción
id	varchar (128)	
title	varchar (500)	Atributo que permite guardar el título del sco al que hace referencia.
description	text	Atributo que permite guardar el campo donde se almacena una breve descripción del sco.
coverage	varchar (128)	Atributo que permite guardar la materia a la cual el sco pertenece.
structure	varchar (50)	Atributo que permite guardar la estructura interna del material. LOM define el siguiente vocabulario controlado para describir la estructura: <i>collection</i> (colección), <i>mixed</i> (mixta), <i>linear</i> (lineal), <i>hierachical</i> (jerárquica), <i>networked</i> (en red), <i>branched</i> (ramificada), <i>parceled</i> (compartimentada), <i>atomic</i> (atómica).
aggregation_level	numeric (9,0)	Atributo que permite guardar la granularidad del material
created_at	timestamp	Atributo que permite guardar la fecha de creación del sco.
updated_at	timestamp	Atributo que permite guardar la fecha de la última modificación realizada al sco.
deleted_at	timestamp	Atributo que permite guardar la fecha de eliminación del sco.

Tabla 3.10 Descripción de la tabla tb_general.

tb_template		
Descripción: En esta tabla se guardan los distintos tipos de plantillas que se manejan en la creación de las páginas.		
Atributo	Tipo	Descripción
id	VARCHAR(128)	
sco_type_id	VARCHAR(128)	Atributo que permite guardar el identificador del tipo de sco.
small_picture	text	Atributo que permite guardar una imagen serializada.
title	VARCHAR(128)	Atributo que permite guardar el título de la plantilla.
description	VARCHAR(500)	Atributo que permite guardar una breve descripción de la plantilla.
sco_text	text	Atributo que permite guardar el contenido de las plantillas.
replication	varying(5)	Atributo que permite guardar los bits de replicación.

CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA

created_at	timestamp	Atributo que permite guardar la fecha de creación de la plantilla.
updated_at	timestamp	Atributo que permite guardar la fecha de la última modificación realizada a la plantilla.
deleted_at	timestamp	Atributo que permite guardar la fecha de eliminación de la plantilla.

Tabla 3.11 Descripción de la tabla tb_template.

tb_link		
Descripción: En esta tabla se guardan los recursos de tipo Link.		
Atributo	Tipo	Descripción
id	VARCHAR(128)	
link	varying(1000)	Atributo que permite guardar una url a un sitio de interés.
description	text	Atributo que permite guardar una breve descripción del recurso.
replication	varying(5)	Atributo que permite guardar los bits de replicación.
created_at	timestamp	Atributo que permite guardar la fecha de creación del recurso Link.
updated_at	timestamp	Atributo que permite guardar la fecha de la última modificación realizada al recurso.
deleted_at	timestamp	Atributo que permite guardar la fecha de eliminación del recurso.

Tabla 3.13 Descripción de la tabla tb_link.

r_glosary_matter		
Descripción: En esta tabla se guardan las relaciones entre los términos y sus significados en las diferentes materias.		
Atributo	Tipo	Descripción
id	VARCHAR(128)	
matter_id	VARCHAR(128)	Atributo que permite guardar la palabra del glosario.
glosary_id	VARCHAR(128)	Atributo que permite guardar el identificador del término.
description	text	Atributo que permite guardar la descripción asociada al término en esa materia.
replication	varying(5)	Atributo que permite guardar los bits de replicación.
created_at	timestamp	Atributo que permite guardar la fecha de creación de la relación.
updated_at	timestamp	Atributo que permite guardar la fecha de la última modificación realizada a la relación.
deleted_at	timestamp	Atributo que permite guardar la fecha de eliminación de la relación.

Tabla 3.18 Descripción de la tabla r_glosary_matter.

nom_language		
Descripción: En esta tabla se guardan los recursos de tipo ejercicios.		
Atributo	Tipo	Descripción

CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA

id	VARCHAR(128)	
name	VARCHAR(128)	Atributo que permite guardar el nombre del idioma.
language	VARCHAR(128)	Atributo que permite guardar el lenguaje.
type	VARCHAR(128)	Atributo que permite guardar el tipo de idioma.
description	text	Atributo que permite guardar una breve descripción del nomenclador de lenguajes.
replication	varying(5)	Atributo que permite guardar los bits de replicación.
created_at	timestamp	Atributo que permite guardar la fecha de creación del nomenclador de lenguajes.
updated_at	timestamp	Atributo que permite guardar la fecha de la última modificación del nomenclador de lenguajes.
deleted_at	timestamp	Atributo que permite guardar la fecha de eliminación del nomenclador de lenguajes.

Tabla 3.20 Descripción de la tabla nom_language.

3.6. Conclusiones parciales

En este capítulo quedó definido el análisis y diseño del sistema así como las relaciones definidas en este, describiendo estas en diagramas de clases, enfocados al cumplimiento de los requerimientos funcionales del componente. Se realizó una valoración de los patrones de diseño teniendo en cuenta las características del componente valorando la aplicación del modelo arquitectónico con vista a dichas características. Se obtuvo el diseño de la base de datos lo cual unido a todo lo obtenido en el capítulo proporciona las bases para la implementación del componente.

IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

El capítulo recoge toda la documentación del proceso de implementación de las funcionalidades identificadas en la descripción del diseño. Los artefactos que se modelan son el diagrama de despliegue y el diagrama de componentes. También se recogen los resultados de las pruebas realizadas a los diferentes componentes.

4.2 Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño, así como las clases se implementan en términos de componentes, en forma de ficheros de código fuente, ejecutables, entre otros. El mismo describe además como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros [21].

Este modelo representa la composición física de la implementación en término de subsistemas de implementación, y elementos de implementación (directorios y archivos incluyendo código fuente, datos y archivos ejecutables). Además, define las principales unidades de integración alrededor de las cuales se organizan los equipos, así como las unidades que se pueden versionar, desplegar y reemplazar separadamente.

4.2.1 Diagrama de Despliegue

El diagrama de despliegue define la arquitectura física que presenta el sistema. Se usa para modelar detalladamente los nodos físicos y las comunicaciones existentes entre ellos. De ese mismo modo queda especificado qué hardware, sistema operativo, software de interfaces y soporte conformarán los componentes implementados.

A continuación se muestra el diagrama de despliegue propuesto para el sistema:

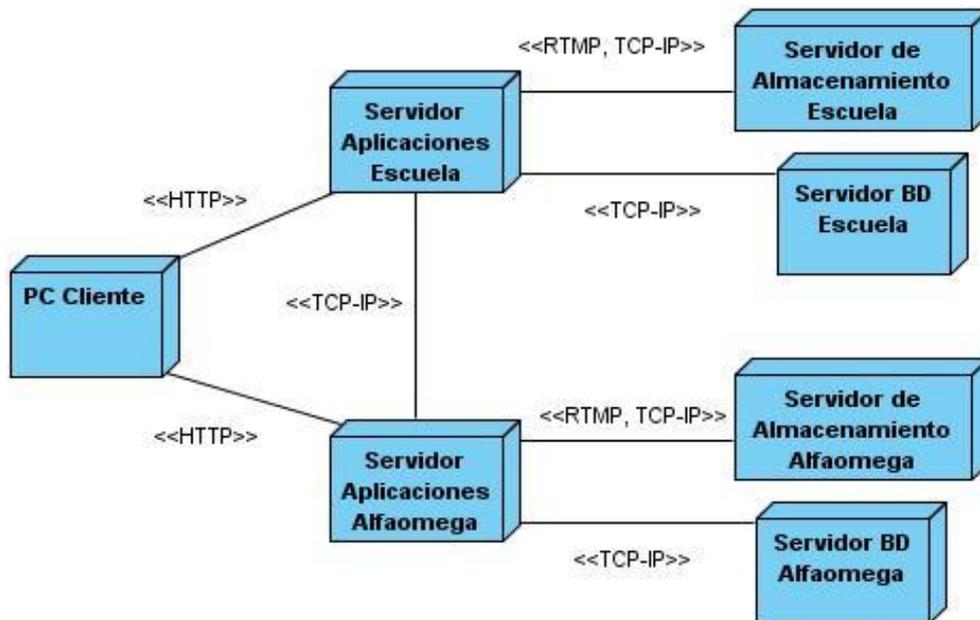


Figura 4.1 Diagrama de Despliegue.

4.2.2 Diagrama de Componentes

El diagrama de componentes se utiliza para que los usuarios que utilicen el producto final puedan ver la estructura del sistema finalizado, a su vez posibilitarle a los desarrolladores una estructura detallada del modelo de implementación que les permita poder entender el sistema y en caso de ser necesario mejorarlo.

El uso más importante de este diagrama es mostrar la estructura de alto nivel de modelo de implementación especificando de manera detallada las organizaciones y dependencias entre los diferentes componentes, así como la organización de implementación en las diferentes capas.

A continuación se muestra el diagrama de componentes propuesto para el sistema:

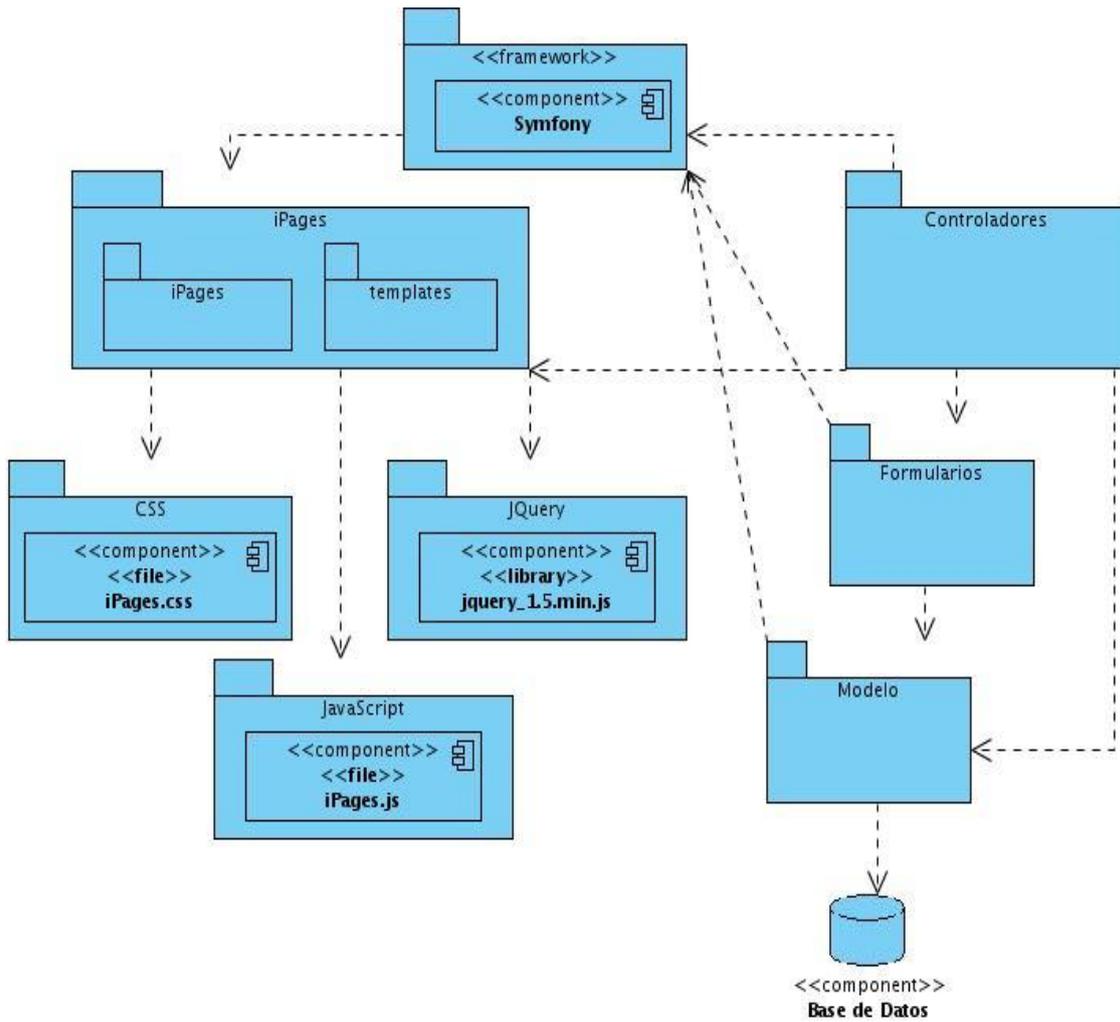


Figura 4.2 Diagrama de Componentes.

Seguido del anterior diagrama de componentes se mostrarán las descripciones de los siguientes paquetes incluidos en el mismo, a continuación se tiene la descripción de los paquetes Vistas, Controladores, Modelo y Formularios:

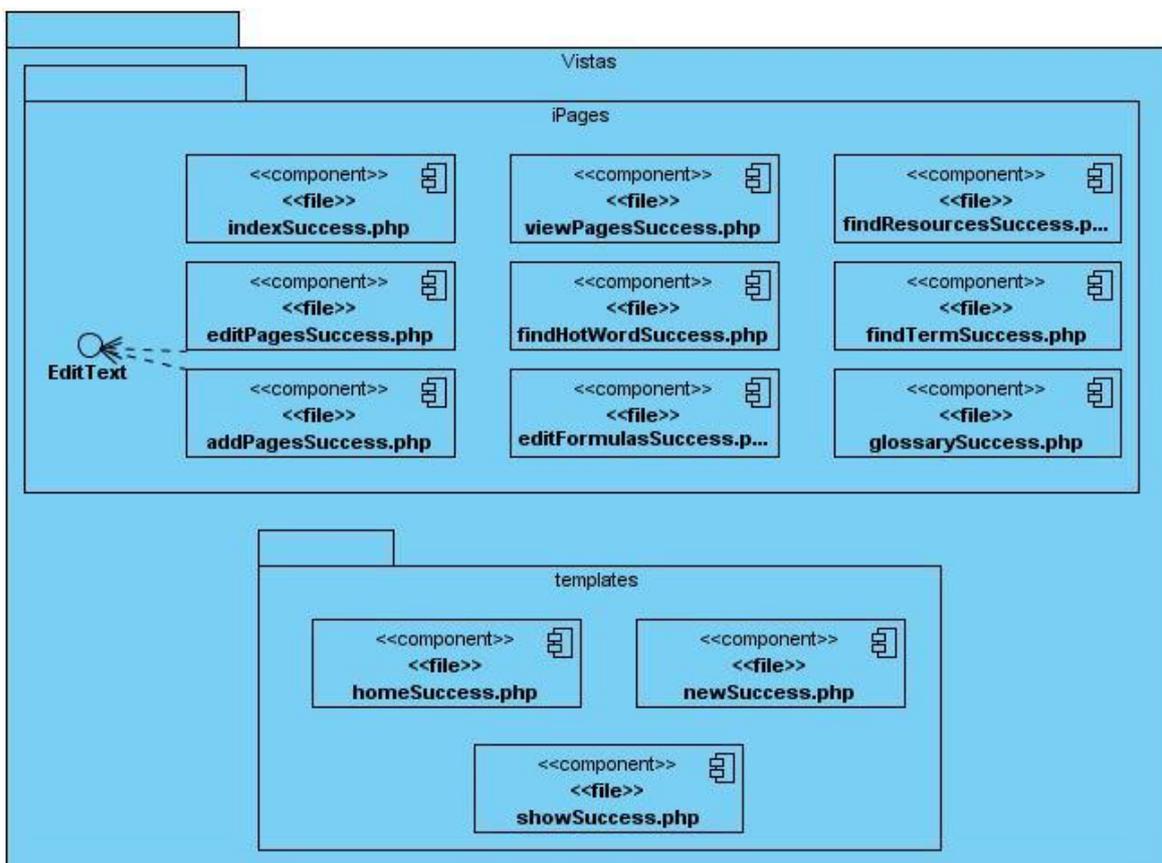


Figura 4.3 Descripción del Paquete Vistas.

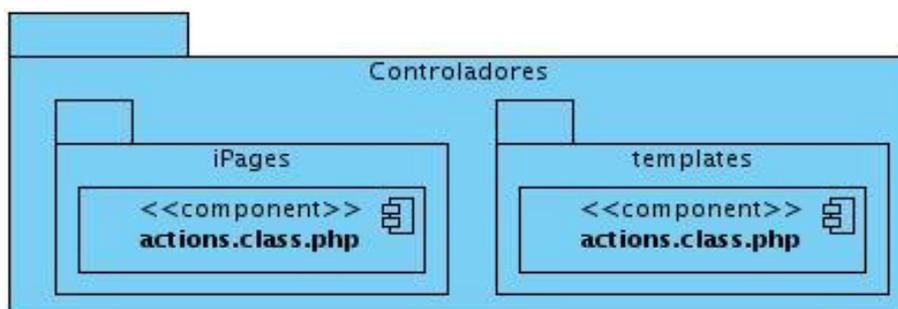


Figura 4.4 Descripción del Paquete Controladores.

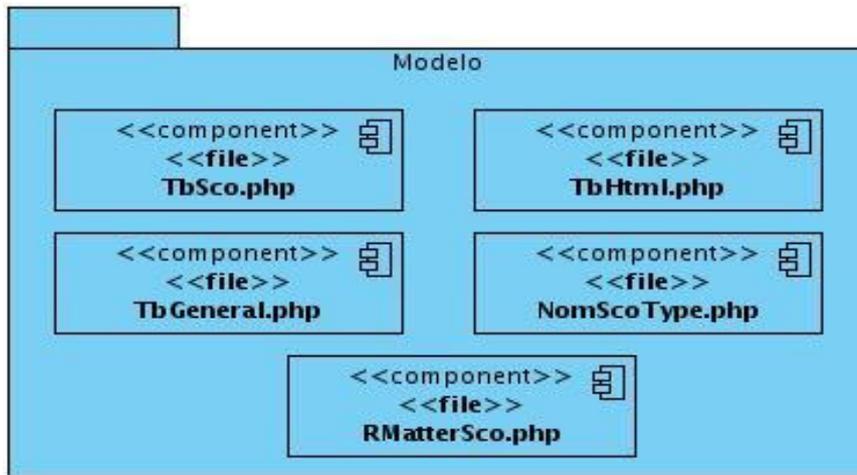


Figura 4.5 Descripción del Paquete Modelo.

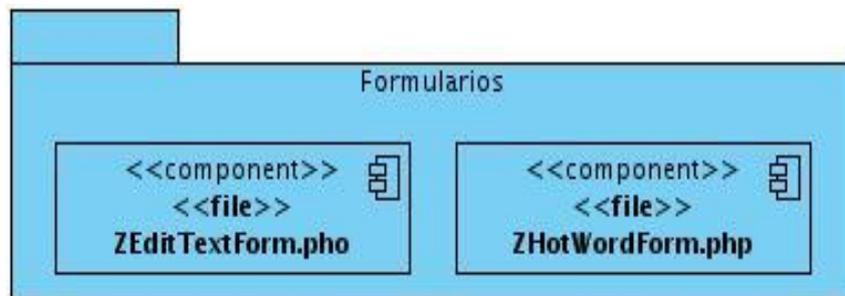


Figura 4.6 Descripción del Paquete Formularios.

4.3 Pruebas de Software

Las pruebas de software son un elemento crítico para la garantía de la calidad de software, su objetivo principal es garantizar la calidad del producto desarrollado, esta etapa implica verificar la iteración de los componentes y su adecuada integración; verificar la correcta implementación de todos los requisitos e identificar y asegurar que los defectos encontrados se han corregido antes de entregar el producto.

4.3.1 Niveles de Prueba

El proceso de pruebas pasa por diferentes niveles los cuales se implementan de manera secuencial relacionándose con los productos de desarrollos estos niveles son:

- ❖ **Prueba de Unidad:** esta se centra en el módulo haciendo uso de la descripción del diseño detallado como guía, se prueban los caminos de control importantes con el fin de descubrir errores dentro del ámbito del módulo. La prueba de unidad se basa principalmente en las técnicas de prueba de caja blanca.

- ❖ **Prueba de Integración:** su objetivo principal es utilizar los módulos probados en el nivel anterior y construir una estructura de programa que esté de acuerdo con lo que especifica el diseño.

Existen dos maneras de realizar la integración, estas son:

- *Integración no incremental:* Se combinan todos los módulos por anticipado y se prueba todo el programa en conjunto.
 - *Integración incremental:* El programa se construye y se prueba en pequeños segmentos. En la prueba de integración el foco de atención es el diseño y la construcción de la arquitectura del software.
- ❖ **Prueba de Validación:** en esta etapa el software totalmente ensamblado se prueba como un todo para comprobar si cumple los requisitos funcionales y de rendimiento, facilidad de mantenimiento, recuperación de errores, etc.
 - ❖ **Prueba del Sistema:** el software ya validado se integra con el resto del sistema (rendimiento, seguridad, recuperación y resistencia)
 - ❖ **Prueba de aceptación:** el usuario comprueba en su propio entorno de explotación si lo acepta o no en su fase final.

4.3.2 Método de Prueba

❖ Pruebas de Caja Negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, y es completamente indiferente al comportamiento interno y la estructura del programa. Estas pruebas se realizan con el objetivo detectar errores tales como funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación.

4.3.2.1 Diseño de Casos de Prueba

El propósito de un caso de prueba es identificar y comunicar las condiciones que se llevarán a cabo en la prueba. Los casos de la prueba son necesarios para verificar la aplicación exitosa y aceptable de los requisitos del producto (casos de uso). A continuación se muestra la descripción del CP Gestionar Página; el resto puede encontrarse en el **Anexo # 6: Descripción de los Casos de Prueba del Sistema.**

Descripción del CP Gestionar Página.

Descripción general

El caso de uso se inicia cuando el Gestor de Páginas selecciona la opción que le permite realizar una acción sobre la página. El actor puede incluir, ver, modificar y eliminar una página. En caso de que seleccione la opción de incluir página, el sistema dará la posibilidad de insertar los elementos y los datos que el actor desee para la confección de su página. Si el actor elige la opción de ver página el sistema mostrará una vista previa del contenido de la página en cuestión. Si el actor elige la opción de modificar página, el sistema mostrará los datos que pueden ser editables dentro de la página, y una vez realizados los cambios los actualiza dentro de ella. Si el actor elige la opción de eliminar una página, el sistema mostrará los datos y permitirá eliminar la misma.

Condiciones de ejecución

Debe haberse generado el escritorio de trabajo del usuario autenticado.

Debe haberse seleccionado la materia.

Debe haberse seleccionado el índice.

Para incluir una página, el subtema asociado debe estar seleccionado previamente.

Para ver una página, debe estar seleccionada previamente, y el actor debe tener permiso de acceder al contenido, ya sea porque es el autor, un rol superior en jerarquía, o porque se le ha asignado temporalmente.

Para modificar una página, debe estar seleccionada previamente y el actor debe ser el responsable temporal de la misma.

Para eliminar una página, debe estar seleccionada previamente y el actor debe tener el permiso de eliminar el elemento.

SC 1 Incluir página.

ID del escenario	Escenario	Respuesta del sistema	Flujo Central
EC 1	Incluir una página	Brinda la posibilidad de seleccionar una de las plantillas creadas por el usuario.	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en el índice donde se desea insertar la página nueva/ Incluir Página.
EC 2	Selecciona la plantilla.	Permite editar el texto dentro de la plantilla seleccionada. Y brinda la posibilidad de asociar un recurso a la plantilla, dependiendo del tipo de esta.	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en el índice donde se desea insertar la página nueva/ Incluir Página/ Clic en una de las plantillas que salen en la parte derecha/ Clic en la opción editar texto dentro de la plantilla
EC 3	Selecciona la opción de asociar un recurso a la	Muestra todos los recursos del tipo seleccionado asociados al índice de la materia.	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en el índice donde se desea insertar

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBA

	plantilla seleccionada.	<p>Y brinda la posibilidad de seleccionar el recurso y editar la vista previa del mismo de acuerdo a:</p> <ul style="list-style-type: none"> • Ancho. • Alto. • Borde. • Color de borde. <p>Y permite además:</p> <ul style="list-style-type: none"> • Cambiar el recurso asociado a la plantilla. • Incluir una palabra o frase como caliente o clave en la plantilla. • Seleccionar una nueva plantilla. • Selecciona editar el texto dentro de una plantilla. • Guardar la página. • Guardar la página sin publicar. • Guardar la página y continuar, en cualquier momento. • Cancelar la operación en cualquier momento. 	la página nueva/ Incluir Página/ Clic en una de las plantillas que salen en la parte derecha del escritorio de trabajo/ Clic en la opción editar recurso dentro del área de la plantilla/ Selecciona una imagen a mostrar en la vista previa de la plantilla es decir el recurso que va a mostrar/ Asociar recursos a la página mediante los botones para esto.
EC 4	Selecciona la opción de guardar la página.	<p>Valida los datos. <i>Crea la página y publica su contenido.</i> Muestra un mensaje de información.</p>	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en el índice donde se desea insertar la página nueva/ Incluir Página/Botón Guardar.
EC 5	Selecciona la opción de cancelar.	<p>Elimina los datos creados. Muestra un mensaje de información. Regresa a la vista anterior.</p>	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en el índice donde se desea insertar la página nueva/ Incluir Página/ Botón Cancelar.
EC 6	Selecciona la opción de Guardar y continuar.	<p>Valida los datos. <i>Guarda los datos de la página y se mantiene en la misma para seguir editándola.</i> Regresa al EC 1.</p>	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en el índice donde se desea insertar la página nueva/ Incluir Página/ Botón Guardar y continuar.
EC 7	Selecciona la opción de	<p>Valida los datos. <i>Crea la página pero no publica su</i></p>	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en el

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBA

	Guardar sin publicar.	<i>contenido.</i> Muestra un mensaje de información. Permite incluir una nueva página.	índice donde se desea insertar la página nueva/ Incluir Página/ Botón Guardar sin publicar.
EC 8	Selecciona la opción de seleccionar una nueva plantilla.	Regresa al EC 1 .	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en el índice donde se desea insertar la página nueva/ Incluir Página/ Clic en una de las plantillas que salen en la parte derecha del escritorio de trabajo.
EC 9	Selecciona la opción de incluir palabra o frase como caliente o clave.	<u>Ver CP Gestionar Palabra Caliente o Frase.</u> Regresa al EC 3 .	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en el índice donde se desea insertar la página nueva/ Incluir Página/ Clic en una de las plantillas que salen en la parte derecha del escritorio de trabajo/ Clic en la opción editar texto dentro de la plantilla/ Marcar la palabra que se desea como clave/ Seleccionar el ícono <i>Crear palabra caliente</i> .
EC 10	Selecciona la opción de editar texto dentro de una plantilla.	<u>Ver CP Editar Texto.</u> Regresa al EC 3 .	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en el índice donde se desea insertar la página nueva/ Incluir Página/ Clic en una de las plantillas que salen en la parte derecha del escritorio de trabajo/ Clic en la opción editar texto dentro de la plantilla.

SC 2 Vista previa de la página.

ID del escenario	Escenario	Respuesta del sistema	Flujo central
EC 1	Ver una vista previa de la página.	Muestra la página según los estilos del hiperentorno a la cual está asociada. Y permite: <ul style="list-style-type: none"> • Salir de la vista actual. 	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Página/ Clic en la página para ver la vista previa.
EC 2	Selecciona la opción de salir de la vista actual.	Muestra la vista anterior.	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Página/ Clic en la página para ver la vista previa/ Botón Salir.

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBA

SC 3 Modificar contenido de la página.

ID del escenario	Escenario	Respuesta del sistema	Flujo central
EC 1	Modificar página.	Muestra el contenido de la página y brinda la posibilidad de cambiarlo. Permite además: <ul style="list-style-type: none"> • Actualizar los cambios. • Actualizar sin publicar los cambios. • Cancelar la operación en cualquier momento. 	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en la página que se desea modificar/ Modificar.
EC 2	Selecciona la opción de actualizar los cambios.	Muestra un mensaje de información y muestra la vista previa de la página Ver SC 2 del CP Gestionar Página. Publica el contenido de la página.	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en la página que se desea modificar/ Modificar/ Botón Actualizar.
EC 3	Selecciona la opción de actualizar sin publicar los cambios.	Muestra un mensaje de información y muestra la vista previa de la página Ver SC 2 del CP Gestionar Página. No publica el contenido de la página.	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en la página que se desea modificar/ Modificar/ Botón Actualizar sin publicar.
EC 4	Selecciona la opción de cancelar.	Muestra un mensaje de información y regresa a la vista anterior.	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en la página que se desea modificar/ Modificar/ Botón Cancelar.

SC 3 Eliminar página.

ID del escenario	Escenario	Respuesta del sistema	Flujo central
EC 1	Eliminar página.	Muestra un mensaje de advertencia.	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en la página que se desea eliminar/ Eliminar.
EC 2	Selecciona la opción de aceptar.	Oculto la página. Muestra un mensaje de información. Regresa a la vista anterior actualizando los datos.	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en la página que se desea eliminar/ Eliminar/ Botón Aceptar.

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBA

EC 3	Selecciona la opción de cancelar.	Regresa a la vista anterior.	Escritorio de trabajo/ Recursos Interactivos/ Páginas Interactivas/ Clic derecho en la página que se desea eliminar/ Eliminar/ Botón Cancelar.
-------------	-----------------------------------	------------------------------	--

Descripción de las variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Listado de recursos	Lista desplegable.	No	Nombre (s) del recurso a asociar a la página.
2	Ícono	Campo de selección.	No	Para seleccionar como se va a mostrar el recurso en la página.
3	Imagen	Campo de selección.	No	Para seleccionar como se va a mostrar el recurso en la página.

4.4 Resultados Obtenidos

Tipos No Conformidades	1era Iteración	2da Iteración
Formato (A y D)		
Error Técnico (D)		
Correspondencia con otro artefacto (A y D)	5	
Ortografía (A y D)		
Redacción (A y D)		
Validación (A)		
Opciones que no funcionan (A)		
Error de Interfaz (A)		
Error Idioma (A)	1	
Funcionalidad (A)	1	
Excepciones (A)		

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBA

Otros (A y D)		
Seguridad (A)		
Diseño CP (D)	4	
Recomendación (A y D)		1
Total	11	1

Tabla 4.1 Resultados de las Pruebas de Liberación.

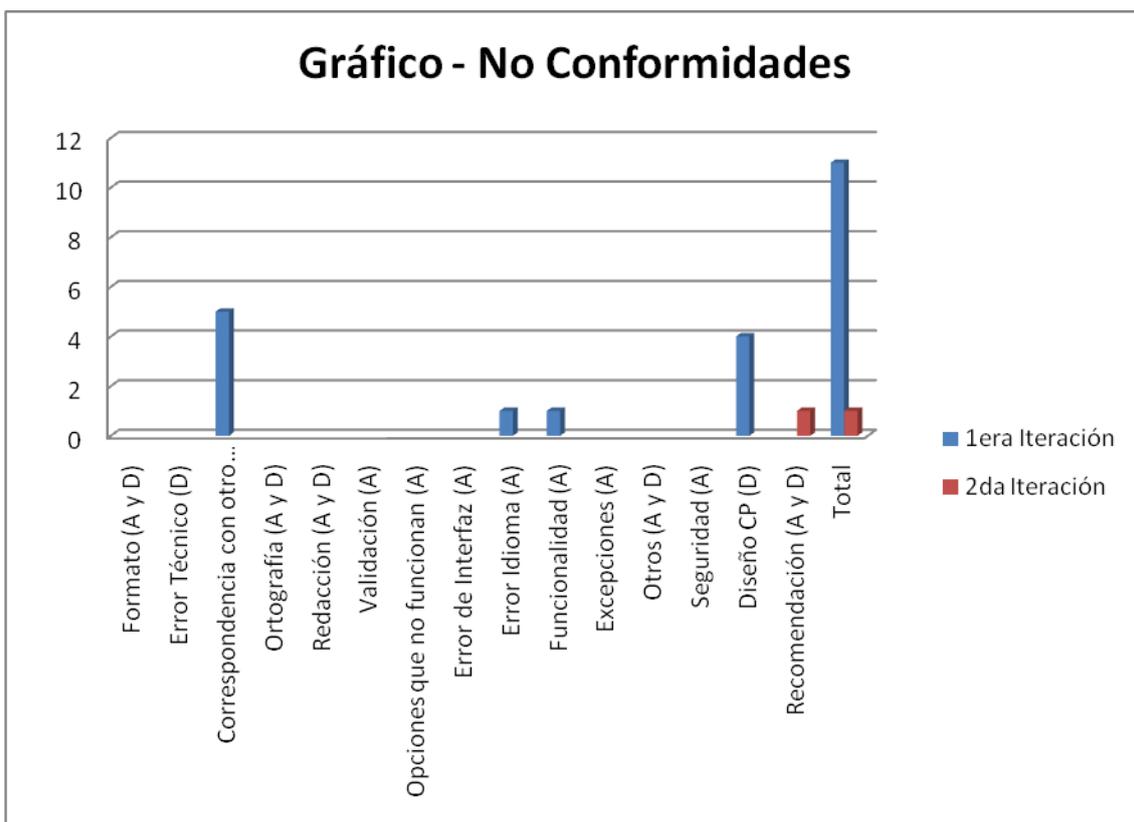


Figura 4.7 Gráfico de los resultados de las Pruebas de Liberación.

4.5 Conclusiones

En este capítulo se relacionaron los principales elementos de la implementación de los componentes desarrollados. Se realizó el diagrama de despliegue para con ello lograr una representación estructural del sistema en general; también se obtuvo el diagrama de componentes para que los usuarios finales que utilicen el producto puedan observar la estructura del mismo finalizado y con ello posibilitarle también a los desarrolladores una estructura detallada del modelo de implementación que permita que entiendan el sistema y puedan hacerle cambios para futuras versiones del mismo.

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBA

Se realizó el proceso de pruebas en la fase de pruebas de liberación para ello se llevó a cabo el diseño de casos de prueba definido por el método de pruebas de caja negra, dichas pruebas arrojaron resultados que permitieron que se le corrigieran los errores detectados al sistema y se obtuviera el mismo con una mayor calidad.

CONCLUSIONES

Con la realización del presente trabajo de diploma se obtuvo como resultado la implementación de componentes que permiten la gestión de las páginas de forma estructurada e interactiva, se creó un componente para la gestión de las plantillas que el usuario desee predefinir para estructurar el contenido dentro de sus páginas de la manera que desee; se obtuvo la documentación propuesta por los flujos de trabajo de la metodología utilizada en este caso Proceso Unificado de Desarrollo (RUP) con ello se demostró lo beneficioso que resulta la utilización de una metodología tan sólida como RUP paralelamente a la utilización del framework Symfony, lo que propició la implementación efectiva de la aplicación. , entre ellos el diagrama de clases del diseño el cual da una vista del sistema en forma de paquetes; también se muestra el diagrama de componentes y el de despliegue pertenecientes al modelo de implementación lo cual permitirá que en futuras versiones del sistema los desarrolladores puedan entender la lógica de los componentes implementados.

RECOMENDACIONES

Luego de la culminación del trabajo se recomienda lo siguiente:

- ❖ Realizar algunas modificaciones en los componentes, para que no solo puedan ser utilizados en la plataforma ZERA, sino que pueda ser reutilizable por otros proyectos de la Universidad que así lo requieran.
- ❖ Lograr una estrategia que disminuya el tiempo en la búsqueda de los recursos en la plataforma, debido a que cuando el volumen de recursos aumente notablemente como se espera que ocurra en el futuro, el proceso actual podría tornarse un poco lento .
- ❖ Implementar nuevas funcionalidades en versiones posteriores.
- ❖ Mejorar el diseño de interfaz de usuario, para mejorar la apariencia visual.

BIBLIOGRAFÍA

1. **Márquez, P.** El software educativo. [Online]
http://www.lmi.ub.es/te/any96/marques_software/#capitol1.
2. **Agudelo, M. M.** Plataformas educativas. [Online] 2006. [Cited: 06 25, 2010.]
<http://aprendeonline.udea.edu.co/banco/html/plataformaseducativas/>.
3. **LCMS Y OBJETOS DE APRENDIZAJE. Vázquez, Larisa Enríquez.** 2004.
4. **MODELO DE DISEÑO DIDÁCTICO DE HIPERENTORNOS DE ENSEÑANZA– APRENDIZAJE DESDE UNA CONCEPCIÓN DESARROLLADORA. Rodríguez, MsC. Mario del Toro.** Ciudad de La Habana : s.n., 2006.
5. **Administración, Manejo y.** WordPress.org. [Online] [Cited: Enero 12, 2011.]
http://codex.wordpress.org/es:WordPress_Features.
6. **Claroline, Consorcio.** Claroline.net. [Online] 2008. [Cited: Enero 11, 2011.]
<http://www.claroline.net/index.php?lang=es>.
7. **Manjón, Baltasar Fernández.** Ministerio de Educación y Ciencia. [Online] [Cited: Enero 11, 2011.] <http://ares.cnice.mec.es/informes/16/contenido/47.htm>.
8. **Navarra, Universidad Pública de.** [Online] 2009. [Cited: Enero 11, 2011.]
<http://www.navarrainnova.com/es/actualidad/notas-de-prensa/2009/05/03/15417.php>.
9. **Escribano, Gerardo Fernández.** [Online] Diciembre 9, 2002. [Cited: Enero 11, 2011.] <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.
10. **Pérez, Javier Eguíluz.** Introducción a XHTML. [Online] 2008. [Cited: Enero 11, 2011.] <http://librosweb.es/xhtml/index.html>.
11. —. Introducción a CSS. [Online] 2008. [Cited: Enero 11, 2011.]
<http://librosweb.es/css/index.html>.
12. **Hinostroza, Raul Rodas.** LinuxCentro.net. [Online] Febrero 22, 2007. [Cited: Enero 11, 2011.]
<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
13. **Paradigm, V.** Visual Paradigm for UML. [Online] 2010.
http://translate.googleusercontent.com/translate_c?hl=es&sl=en&u=http://www.visual-paradigm.com/product/vpuml/&prev=/search%3Fq%3Dvisual%2Bparadigm%26hl%3Des&rurl=translate.google.com.cu&usg=ALkJrhjwX7grUD9526vtJznJGro1ys5X2w.
14. **Group, PostgreSQL Global Development.** PostgreSQL. [Online] 2010.
www.postgresql.org.
15. **Guerrero, S.** Netbeans 6.8 con soporte para PHP 5.3 y Symfony. [Online] 2009.
<http://www.sergioguerrero.es/php/netbeans-soporte-php53-symfony/>.
16. **Hinostroza, Raul Rodas.** LinuxCentro.net. [Online] 2007.
<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
17. **Gutiérrez, Javier.** ¿Qué es un framework web? [Online] 2009. [Cited: Enero 11, 2011.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
18. **Saltares, David.** [Online] 8 1, 2010. <http://siondream.com/blog/desarrollo-informatica/pencil-project-bocetos-de-interfaz/>.
19. **symfony.es.** [Online] 2011. [Cited: 3 21, 2011.]
<http://www.symfony.es/categoria/propel/>.
20. **Jacobson, Ivan, Booch, Grady and Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Pearson Educación.S.A, 2000.
21. **jQuery.** The jQuery Project. [Online] 2010. <http://jquery.com/>.
22. **Template SAGA - Manual FCKEditor.** [Online] [Cited: 4 13, 2011.]
<http://www.templatesaga.com/nav/documentacion/editores/manualFCKEditor.html>.
23. **Knabben, CKSource - Frederico.** CKEditor. [Online] 2003 - 2011. [Cited: 3 24, 2011.] <http://ckeditor.com/%5D>.

24. **Oficial, Sitio.** tinymce. [Online] [Cited: 3 25, 2011.]
<http://tinymce.moxiecode.com/index.php>.
25. **Eleven, Sitio Oficial;** Eleven, e_learning virtual environment. [Online] 2010.
<http://www.plataformaeleven.com/es/platform.html>.
26. **Pérez, J. E.** Introducción a JavaScript. [Online] 2009. [Cited: 5 15, 2010.]
<http://librosweb.es/>.

GLOSARIO DE TÉRMINOS

Software Educativo: Software destinado a la enseñanza y el aprendizaje individual o independiente y que además, permite el desarrollo de ciertas habilidades cognitivas.

Hiperentorno de Aprendizaje: se define como una modalidad informática que se sustenta en la tecnología hipertexto y en el que están presentes un conjunto de elementos representativos de diversas tipologías de software educativo.

WYSIWYG: es el acrónimo de What You See Is What You Get ("lo que ves es lo que obtienes")

Metodología de Desarrollo: es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Framework: se traduce en una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

ORM (Object Relation Mapper o Mapeo de Objetos Relacional): es una técnica de programación que brinda la posibilidad de convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

Página: Conjunto de contenidos agrupados de acuerdo a una materia dada.

Plantilla: Es un elemento que puede ser creado por el usuario a su gusto para estructurar el contenido dentro de la página como tal.

Recurso multimedia: Se entenderá por recursos multimedia la información presentada a partir de cualquiera de los siguientes 5 formatos digitales o sus combinaciones: Texto o hipertexto, Video digital, Sonido digital, Animaciones e Imagen fija.

Recurso estructural: Se trata de un conjunto de recursos cuya función es la de auspiciar un andamiaje de apoyos didáctico que posibilite establecer elementos del contenido con marcada intencionalidad pedagógica.

Recurso interactivo: Sustentado en el modelo de comunicación Shannon Weber, por interactividad en el marco informático, se entiende tanto la interactividad hombre-máquina

como a que se produce entre hombres vehiculada por la máquina (hombre-hombre mediada por la máquina).