

Universidad de las Ciencias Informáticas

FACULTAD 6



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

Título: Aplicación para la Generación de Libros Web v2.0.

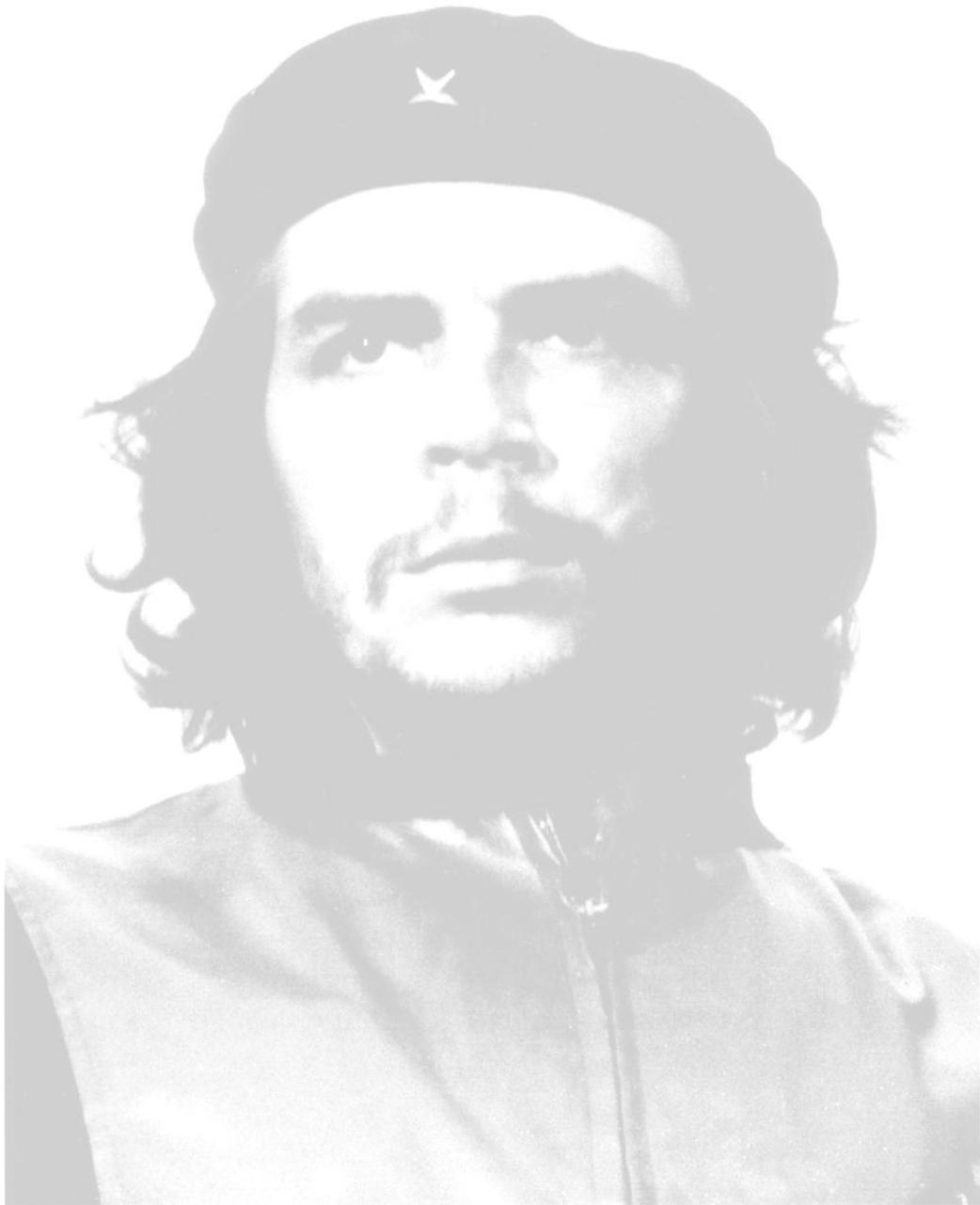
Autor: Alejandro González Sánchez.

Tutor: Ing. Virgilio Suárez Bello.

Ing. Claudia Núñez Sanz.

La Habana, Junio de 2013

“Año 55 de la Revolución”



“La responsabilidad nuestra es luchar porque la calidad del producto que aquí se haga sea de las mejores y la mejor posible...”

Ernesto Che Guevara

DECLARACIÓN DE AUTORÍA

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

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

Alejandro González Sánchez

Firma del Autor

Virgilio Suárez Bello

Firma del Tutor

Claudia Núñez Sans

Firma del Tutor

DATOS DE CONTACTO

Autor:

Alejandro González Sánchez

Universidad de las Ciencias Informáticas

La Habana, Cuba

E-mail: alejandrog@estudiantes.uci.cu

Tutores:

Ing. Virgilio Suárez Bello

Universidad de las Ciencias Informáticas

La Habana, Cuba

E-mail: vsuarez@uci.cu

Ing. Claudia Núñez Sanz

Universidad de las Ciencias Informáticas

La Habana, Cuba

E-mail: cnunez@uci.cu

AGRADECIMIENTOS

Muchas han sido las personas que a lo largo de estos cinco años, han contribuido a que hoy me encuentre aquí. Este es el momento de agradecer a cada una de ellas.

Primeramente quiero agradecer a mi mamá Rosario y a mi papá Arturo por su amor y cariño, por apoyar incondicionalmente todas mis decisiones y por el esfuerzo y trabajo con el cual me han educado, haciendo de mí la persona que soy actualmente.

A mi hermano Amaury, por siempre estar a mi lado y ser mi motor impulsor ante cualquier adversidad.

A mi novia Diana, por enseñarme a ver las cosas desde otra perspectiva y por ser incondicional a mí y a nuestra relación.

A mi inseparable amigo, mentor y hermano Osvaldo por todos sus consejos y apoyo desde el primer día en la UCI, hasta los días de hoy.

A mi tutor Virgilio Suárez Bello, por su decisivo, estratégico e incondicional apoyo en los momentos más críticos, demostrándome que se podía aunque la marea apuntara a lo contrario.

A todo ese gran equipo de la FEU, con el cual tuve el honor de trabajar y dirigir en los 2 últimos años. Hoy puedo decir que me gradúo de una escuela más, la de la FEU... gracias por su dedicación y consagración, les puedo asegurar que ningún equipo ha hecho tanto en tan poco tiempo...

A todos mis compañeros con los que viví muchos momentos en estos cinco años de carrera, tanto buenos como malos, en especial quiero mencionar a: Rene Leandro, Liandry Monsteslier, Darlon Antonio, Luis Ángel, Marco Michel, Pablo Enrique y muchos más, tantos, que no cabrían en otras 80 páginas.

Por último agradecer a todos los que de una forma u otra han contribuido a lo largo de estos cinco años a mi preparación profesional y como una mejor persona. De corazón muchísimas gracias.

DEDICATORIA

A mi mamá y a mi papá, por ser mi inspiración y el motivo de mí existir.

A mi hermano por estar siempre conmigo y ser mi ejemplo a seguir.

A mi novia Diana, por todo su amor y comprensión.

RESUMEN

En la actualidad los manuales de ayuda pueden presentarse en distintos formatos, entre los de mayor uso se encuentran el Word y PDF, aunque también resalta el reciente uso de los libros electrónicos. Entre las tendencias actuales de los mencionados se encuentran los Libros Web, los cuales incorporan hojas de estilos en cascadas mediante plantillas y estructuran un documento en formato web. Especialistas del Centro de Tecnologías de Gestión de Datos (DATEC) de la Universidad de las Ciencias Informáticas (UCI), desarrollan el proyecto: Libros Web, que tiene como objetivo principal la generación de los manuales de ayuda en este formato para todos los productos del centro. Esta aplicación en la actualidad tiene un grupo de características que limitan el trabajo con la herramienta, entre estas se encuentra la inexistente opción de modificar el contenido de los Proyectos Libro Web (PLW) luego de ser creados, el sistema no permite exportar los proyectos y a la vez ser importado por determinado usuario, no se cuenta con un módulo estadístico que permita la visualización de los datos que gestiona el sistema, entre otras limitantes. Atendiendo a los aspectos antes mencionados se desarrolló la Aplicación para la Generación de Libros Web 2.0, incorporándole funcionalidades que permitan una mejor gestión y edición de los manuales de ayuda, mediante el trabajo con roles en el propio sistema.

Palabras claves: aplicación web, libros electrónicos, libros web.

ÍNDICE DE CONTENIDOS

AGRADECIMIENTOS	I
DEDICATORIA	II
RESUMEN	III
ÍNDICE DE CONTENIDOS	IV
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	VIII
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTOS TEÓRICOS	5
Introducción.....	5
1.1. Aplicación informática	5
1.2. Aplicación web	6
1.3. Libro electrónico.....	6
1.4. Libro web	7
1.5. Aplicación para la Generación de Libros Web 1.0	7
1.6. Herramientas para el desarrollo de libros electrónicos y libros web.....	8
1.7. Phyton-Sphinx 1.1.3	15
1.8. Metodología de desarrollo	16
1.9. Lenguajes de programación	18
1.10. Marco de trabajo.....	18
1.11. Gestor de bases de datos.....	19
1.12. Entorno de desarrollo	20

1.13. Lenguaje de modelado	21
1.14. Herramienta CASE	21
1.15. Conclusiones parciales del capítulo.....	22
CAPÍTULO 2: ANÁLISIS Y DISEÑO	23
Introducción.....	23
2.1 Modelo de dominio	23
2.2 Descripción del modelo del dominio	23
2.3 Requisitos funcionales	25
2.4 Requisitos no funcionales.....	28
2.5 Diagrama de caso de usos del sistema	32
2.6 Patrones utilizados en la aplicación.....	38
2.7 Diagrama de clases del diseño.....	41
2.8 Diagrama de secuencia.....	43
2.9 Modelo de datos.....	43
2.10 Diagrama de despliegue.....	46
2.11 Conclusiones parciales del capítulo.....	47
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS	49
Introducción.....	49
3.1 Modelo de implementación.....	49
3.2 Diagrama de componentes.....	49
3.3 Estándares de codificación	50
3.3.1 Notación CamelCase	50
3.3.2 Espacios en blancos y declaraciones por línea	51
3.3.3 Ejemplos de códigos fuente.	51

3.4 Pruebas de software.....	52
3.4.1 Niveles de prueba	52
3.4.2 Tipos de prueba	53
3.4.3 Método de prueba	53
3.4.4 Resultados de las pruebas.....	54
3.5 Conclusiones parciales del capítulo.....	56
CONCLUSIONES GENERALES.....	57
RECOMENDACIONES	58
REFERENCIAS BIBLIOGRÁFICAS	59
BIBLIOGRAFÍA.....	62
ANEXOS.....	66
GLOSARIO	68

ÍNDICE DE FIGURAS

Figura 1: Modelo de Dominio	24
Figura 2: Diagrama de Caso de Uso del Sistema (DCUS)	32
Figura 3: Prototipo de Interfaz: Escenario Adicionar Contenido	34
Figura 4: Prototipo de Interfaz: Escenario Adicionar Contenido (Flujo Alternativo)	35
Figura 5: Prototipo de Interfaz: Escenario Actualizar Contenido.....	35
Figura 6: Prototipo de Interfaz: Escenario Actualizar Contenido (Flujo Alternativo)	36
Figura 7: Prototipo de Interfaz: Escenario Eliminar Contenido	37
Figura 8: Prototipo de Interfaz: Escenario Insertar Contenido	37
Figura 9: Prototipo de Interfaz: Escenario Insertar Contenido (Flujo Alternativo)	38
Figura 10: Diagrama de clase del diseño del CU: Gestionar Contenido.....	42
Figura 11: Diagrama de secuencia del escenario eliminar contenido.....	43
Figura 12: Modelo de datos	44
Figura 13: Diagrama de despliegue	47
Figura 14: Diagrama de componentes del CU: Gestionar Contenido	50
Figura 15: Fragmento de código con la notación CamelCase.....	51
Figura 16: Fragmento de código con espacios en blancos.....	51
Figura 17: Fragmento de código con declaraciones por líneas	52
Figura 18: Fragmento de código del CU: Gestionar Contenido	52
Figura 19: Anexo 1: Diagrama de componente del CU: Gestionar Proyecto	66
Figura 20: Anexo 2: Diagrama de clases del diseño del CU Gestionar Proyecto	66
Figura 21: Anexo 3: Interfaz principal de la Aplicación.....	67
Figura 22: Anexo 4: Interfaz principal, funcionalidad abrir proyecto	67

ÍNDICE DE TABLAS

Tabla 1: Características de las herramientas actuales para la generación de libros electrónicos.....	14
Tabla 2: Descripción del caso de uso Gestionar Contenido de PLW.....	38
Tabla 3: Descripción de la Tabla Traza del modelo de datos	45
Tabla 4 Descripción de la Tabla Proyectos del modelo de datos	45
Tabla 5 Descripción de la Tabla Usuarios del modelo de datos	46
Tabla 6: Resultados de las pruebas de rendimiento con la herramienta JMeter.....	55
Tabla 7: Resultados de las pruebas de seguridad.....	55
Tabla 8: Resultados de las pruebas de Caja Negra	56

INTRODUCCIÓN

La informatización de la sociedad se ha convertido en un proceso de evolución constante, partiendo de varias innovaciones que con el propio desarrollo del hombre, fueron creando las bases de la sociedad actual. Entre los primeros pasos se encuentra el surgimiento de la imprenta, la cual introduce un paradigma a nivel mundial basado en la creación de libros, con mejor calidad, mayor cantidad y menos tiempo de espera. Con el progreso de la ciencia y la tecnología surgen los ordenadores, trayendo la nueva concepción de la era digital al mundo, la materialización de la gran red de redes y las bases para compartir todo tipo de información, sin importar situación geográfica e idiosincrasia social.

Con la creciente expansión de la red Internet, varios programas informáticos permiten facilidades como la comunicación instantánea, mediante el correo, chat, video, voz o todas fusionadas. Para un uso más eficiente de toda esta gama de productos surgen los manuales de ayuda, con el objetivo de indicar cómo trabajar no solo con los programas informáticos, sino con equipos electrodomésticos y de otra índole. Los manuales de ayuda pueden presentarse en disímiles formatos, entre estos se encuentran los libros electrónicos.

Los libros electrónicos surgen con el proyecto *Gutenberg* llevado a cabo en la Universidad de Illinois, liderado por *Michael Hart* en 1971. El objetivo principal de este proyecto era crear una biblioteca de libros electrónicos gratuitos, pero, en esta época, la tecnología no estaba suficientemente desarrollada ni estaba al alcance de todos, por lo que fueron necesarios más de veinte años para que el desarrollo tecnológico permitiera llevar a cabo el proyecto. Desde ese momento, el concepto y la forma en que caracterizaba un libro electrónico ha variado, adaptándose a diferentes tendencias y formatos, entre lo que se encuentran los libros web.

A partir del desarrollo de los libros electrónicos los libros web surgen por la necesidad en Internet de mostrar publicaciones que fuesen visualizadas por la mayoría de los internautas, accesibles y compatibles con los navegadores actuales y con aspecto muy similar al de una página *web*. Estas características representan las ventajas que estos poseían frente a otros formatos de libros electrónicos incidiendo positivamente en su difusión en las redes, al punto de ser utilizado fundamentalmente en los sectores literarios y científicos de la sociedad.

El progreso de la informática en Cuba ha conllevado a grandes logros, encaminados estos a propiciar una infraestructura informática capaz de dar respuestas a las actuales necesidades de nuestra nación. Entre estos se encuentra la Universidad de las Ciencias Informáticas (UCI), la cual es creada en septiembre del 2002, con la misión de desarrollar programas, brindar servicios informáticos y desplegar soluciones tecnológicas integrales.

La UCI, cuenta con varios centros de desarrollo, siendo uno de estos el Centro de Tecnologías de Gestión de Datos (DATEC), especializado en proveer soluciones integrales así como servicios informáticos enfocados en el análisis y el almacenamiento de datos. Entre los departamentos pertenecientes a este centro se encuentra Integración de Soluciones, departamento en el cual se desarrolla el proyecto: Libros Web. Este proyecto tiene como objetivo principal visualizar la información de los manuales de usuarios, de los productos liberados en formato libro web. Este cuenta con una Aplicación para la Generación de Libros Web, herramienta que actualmente contiene limitadas características atendiendo a las necesidades reales del departamento.

Entre las limitantes que posee el sistema se encuentra que no permite modificar el contenido de los proyectos luego de ser creados, trayendo consigo que el proceso de confección del libro tenga que realizarse luego de liberado el producto, donde el usuario encargado de crearlo pierde un tiempo considerable que pudiera emplear en otras acciones. No se cuenta con una interfaz visual que se ajuste a la línea de diseño de los productos del centro DATEC, propiciando que el entorno de trabajo no sea lo suficientemente amigable al carecer del uso de imágenes que contrasten con el diseño de los componentes como banner, logo e iconos de las funcionalidades.

No se tiene implementado un módulo que gestione la seguridad con el trabajo de sesiones, roles definidos y proyectos creados por los usuarios, trayendo consigo que sea muy engorroso la asignación de permisos a determinadas acciones por usuarios independientes. De igual manera la aplicación no brinda la opción de exportar e importar proyectos independientemente del usuario que lo haya expedido. El sistema cuenta con limitadas plantillas de diseño predeterminadas y no brinda por roles, opciones asociadas con la gestión de usuarios, contenidos y reasignación de proyectos, donde empleando el manejo de permisos se pueden dividir las responsabilidades en dependencia de las características de cada usuario.

Por la situación problemática planteada con anterioridad se propone en el marco de este trabajo de diploma el siguiente **problema de la investigación**: ¿Cómo contribuir a los proceso de gestión de los proyectos creados en la Aplicación para la Generación de Libros Web 1.0?

Partiendo del problema de la investigación el **objeto de estudio** se enfoca en las aplicaciones y tecnologías para generar libros web definiéndose como **campo de acción** el desarrollo de aplicaciones para la generación de libros web.

Para dar solución al problema de la investigación planteado se propone como **objetivo general** desarrollar la Aplicación para la Generación de Libros Web v 2.0 para el centro DATEC.

El objetivo general abordado con anterioridad se desglosa en los siguientes **objetivos específicos**.

- Establecer los fundamentos teóricos-metodológicos para la generación de libros web.
- Realizar el análisis y el diseño de la Aplicación para la Generación de Libros Web v 2.0.
- Implementar y validar la Aplicación para la Generación de Libros Web v 2.0.

Para dar respuesta a los objetivos planteados se definen las siguientes **tareas de la investigación**:

- Análisis de los conceptos básicos y técnicos implicados en el desarrollo de las aplicaciones generadoras de libros electrónicos y libros web.
- Identificar los requisitos funcionales y no funcionales para el correcto funcionamiento de la aplicación propuesta en su versión 2.0.
- Confeción del modelo de casos de uso del sistema de la aplicación propuesta en su versión 2.0.
- Elaboración del modelo de diseño de la aplicación propuesta en su versión 2.0.
- Elaboración del modelo de implementación de la aplicación propuesta en su versión 2.0.
- Diseño de los casos de prueba definidos a partir de los requisitos funcionales de la aplicación propuesta en su versión 2.0.
- Identificar los tipos de pruebas que se usarán para validar la aplicación en su versión 2.0.

-Documentación y corrección de las no conformidades identificadas en la ejecución de las pruebas.

Posibles resultados:

El aporte práctico de la presente investigación consiste en el desarrollo de la versión 2.0 de la Aplicación para la Generación de Libros Web, la cual permite la edición del contenido de determinado Proyecto Libro Web (PLW) creado por el usuario, exporta e importa un proyecto determinado sin que se encuentre necesariamente en el sistema, incorpora un módulo de gestión estadística, mejora la seguridad mediante el manejo de roles, contiene nuevas plantillas predeterminadas y el ambiente de trabajo se establece acorde a las políticas del centro.

A continuación se realizará una descripción de los 3 capítulos que conformarán el presente trabajo de diploma.

Capítulo 1 Fundamentos Teóricos: Se presenta el estudio del marco teórico de la investigación, mediante un análisis de las principales herramientas, tecnologías y tendencias para generar libros electrónicos así como la Aplicación para la Generación de Libros Web 1.0. Además, se describe y se especifican las tecnologías, herramientas y metodología sobre las cuales se sostendrá el desarrollo posterior.

Capítulo 2 Análisis y Diseño: Se realiza el análisis y diseño de las clases del sistema. Se especifica el modelo de dominio para describir los principales conceptos del negocio. Son representados los requisitos funcionales y no funcionales, así como el diagrama de casos de uso del sistema con sus respectivas descripciones. Se modelan y detallan los diagramas que representan las funcionalidades del sistema y se aplican los patrones de casos de uso, arquitectónicos y de diseño seleccionados para darle cumplimiento al objetivo general.

Capítulo 3 Implementación y Prueba: Se muestra el modelo de implementación como resultado del análisis y diseño anteriormente desarrollado, así como los diagramas de componentes de la aplicación. De igual manera se describe la implementación de la herramienta con sus respectivos estándares de codificación y se validan las garantías de su funcionamiento de acuerdo a las pruebas seleccionadas en las tareas de la investigación.

Capítulo 1: Fundamentos Teóricos

Introducción

En el presente capítulo se aborda todo lo referente a los conceptos básicos que se relacionan con el objeto de estudio y campo de acción de la investigación, partiendo del análisis que tiene como principio definir las herramientas de acuerdo con las características y tecnologías a utilizar para desarrollar la presente investigación. Se brinda una descripción de la metodología, herramientas y tecnologías a utilizar, con el objetivo fundamental de brindar información complementaria para propiciar una mejor comprensión de los capítulos venideros.

1.1. Aplicación informática

Las aplicaciones informáticas son programas diseñados para facilitar al usuario la realización de un determinado tipo de trabajo. Posee ciertas características que le diferencia de un sistema operativo (que hace funcionar al ordenador), de una utilidad (que realiza tareas de mantenimiento o de uso general) y de un lenguaje (con el cual se crean los programas informáticos). Suele resultar una solución informática para la automatización de ciertas tareas complicadas como puede ser la contabilidad o la gestión de un almacén. (1)

Partiendo del concepto abordado con anterioridad el autor del presente trabajo de diploma define que una aplicación informática es un programa de computación destinado a satisfacer un conjunto de tareas determinadas, no es más que un software encaminado a cumplir un propósito en específico donde el usuario tendrá una interacción determinada atendiendo al objetivo de su creación, las mismas pueden estar de manera independiente (Compactador *Winrar*) o en forma de paquete (Paquete de *Microsoft Office*). Estas se pueden clasificar en aplicaciones de escritorio y en aplicaciones web, atendiendo al objeto de estudio del presente trabajo de diploma, en lo adelante se abordará solo las referentes a las aplicaciones web.

1.2. Aplicación web

Una aplicación web se puede definir como una aplicación en la cual un usuario por medio de un navegador realiza peticiones a una aplicación remota accesible a través de Internet (o a través de una red local o mundial) y que recibe una respuesta que se muestra en el propio navegador. (2)

Una aplicación web es básicamente una manera de facilitar el logro de una tarea específica en la web, a diferencia de un sitio web estático que es más bien una herramienta. La aplicación web permite al usuario interactuar directamente con el usuario y los datos, todo en forma personalizada, para llevar a cabo esa tarea específica (3)

Al analizar los fundamentos planteados con anterioridad el autor del presente trabajo de diploma determina que una aplicación web es una aplicación informática que basa su funcionamiento en peticiones a un *host* (servidor) donde está ubicada la aplicación, accediendo mediante un navegador *web*, a diferencia de una aplicación de escritorio que necesita obligatoriamente estar instalada en el ordenador para el trabajo con estas.

1.3. Libro electrónico

Los libros electrónicos pueden definirse como sistemas de información capaces de poner a disposición de sus usuarios una serie de páginas conceptualmente organizadas del mismo modo que las de un libro en formato duro, de esta manera los libros electrónicos no son meras simulaciones de los libros impresos, sino que incluyen propiedades y herramientas que aumentan sus funciones, aprovechando el potencial suministrado por el soporte electrónico. (4)

Los libros electrónicos se clasifican atendiendo al tipo de información que contiene, las funciones que ejecutan y los tipos de metáforas que siguen, entre algunos de estos se encuentran los siguientes:

- ❖ **Libro de Texto:** Consiste en un conjunto de páginas de textos organizadas linealmente.
- ❖ **Libro Parlante:** Contiene páginas de información narradas en audio.
- ❖ **Libro de Imagen en Movimiento:** Contiene animación y material de vídeo en movimiento.
- ❖ **Libro Multimedia:** Combinaciones de texto, sonido, imagen, animación y video.

- ❖ **Libro Hipermedia:** Similares a los libros multimedia pero contienen información que es organizada en una forma no lineal.
- ❖ **Libro Electrónico Inteligente:** Emplean técnicas de inteligencia artificial, como sistemas expertos o redes neuronales, que les permiten aprender acerca de sus usuarios y adapta sus comportamientos en varias formas de manera de alcanzar las necesidades individuales y particulares.
- ❖ **Libro Tele_media:** Requiere el uso de facilidades de las telecomunicaciones para proveer características especiales como la participación del mismo en teleconferencias, el envío de correos electrónicos, y acceso remoto a recursos como a las bibliotecas digitales.
- ❖ **Libro Web:** Integra información organizada en formato web (HTML) con la inclusión de CSS y Java Script.

1.4. Libro web

Libro web es una publicación electrónica cuyo formato es HTML o lenguaje de marcado de hipertexto, constando de varias páginas y vínculos entre ellas. Es posible insertar recursos como imágenes, videos y audio. Debido al avance de las TIC, los libros web también han experimentado cambios como: la forma de organizar su contenido, la inserción de audio y vídeo mediante flash y la inclusión de scripts. (5)

Partiendo de la definición anterior el autor del presente trabajo de diploma define que el libro web es una variante de los libros electrónicos en formato HTML, donde a diferencia de los demás consta de varias páginas con vínculos entre ellas (como un sitio web), los libros web tiene la característica embebida de poseer una estructura jerárquica, que permite la navegación por todo su contenido mediante índices, de igual manera incluyen CSS en sus plantillas e incorporan pequeñas funciones Java Script, características distintivas sobre los demás tipos de libros electrónicos.

1.5. Aplicación para la Generación de Libros Web 1.0

La Aplicación para la Generación de Libros Web es la herramienta fundamental con la que cuenta el Proyecto Libros Web del departamento Integración de Soluciones, con este sistema se confecciona los manuales de ayuda web de los productos realizados por el propio departamento y centro, esta aplicación en la actualidad, tiene las siguientes deficiencias:

- No permite modificar el contenido de los proyectos luego de ser creados.
- No se cuenta con una interfaz visual que se ajuste a la línea de diseño de los productos del centro DATEC.
- No se tiene implementado un módulo que gestione la seguridad con el trabajo de sesiones, roles definidos y proyectos creados por los usuarios, trayendo consigo que sea muy engorroso la asignación de permisos a determinadas acciones por usuarios independientes.
- No permite exportar e importar proyectos independientemente del usuario que lo haya expedido.
- Se cuenta con limitadas plantillas de diseño predeterminadas y no brinda por roles opciones asociadas con la gestión de usuarios, contenidos y reasignación de proyectos, donde empleando el manejo de permisos se pueden dividir las responsabilidades en dependencia de las características de cada usuario.

Como se explica con anterioridad la Aplicación para la Generación de Libros Web 1.0 tiene funcionalidades limitadas atendiendo a las nuevas tendencias sobre los procesos de liberación de los productos y sus respectivos manuales, por tales motivos se decide realizar una versión a la aplicación que permita corregir los aspectos comentados.

1.6. Herramientas para el desarrollo de libros electrónicos y libros web.

A continuación se realizará un análisis de las herramientas que son utilizadas para la generación de libros electrónicos, estableciendo patrones de comparación en dependencia de las características de cada una de ellas y teniendo en cuenta las peculiaridades que darán cumplimiento al objetivo general.

Jutoh

La herramienta Jutoh permite la creación de libros electrónicos en Epub, Mobipocket, formatos de texto Open Document y llano. Jutoh incluye un diseñador de portada sencillo, con plantillas para su uso inmediato, de igual manera se pueden modificar y crear propios diseños con este fin. Jutoh ayuda a importar el contenido existente, ya sea desde un archivo de procesador de texto, archivos HTML o incluso un libro electrónico existente. (6)

Ventajas de la herramienta Jutoh:

- ❖ Rápido para importar el contenido existente, ya sea en texto, HTML u Open Document.

- ❖ Rápido para crear su Epub, Mobipocket u otro formato.
- ❖ Rápido para editar el contenido y cambiar entre las variaciones de sus libros electrónicos.
- ❖ Está escrito en C + + por lo que corre a toda velocidad sin retrasos frustrantes.
- ❖ Es multi-plataforma.
- ❖ Se ejecuta en Windows, Mac, Linux y otros sistemas basados en Unix, y se puede copiar fácilmente sus archivos entre máquinas de diferentes tipos. Una licencia se puede utilizar en múltiples sistemas operativos simultáneamente.

SIGIL

La herramienta Sigil es un editor multi-plataforma que soporta formatos como EPUB y eBook bajo licencia GPLv3. Permite crear libros electrónicos de una manera fácil y cómoda, como en un editor de texto cualquiera, donde también se puede agregar imágenes. Trabaja con WYSIWYG, además da la posibilidad de editar los metadatos, importar archivos de texto, HTML y EPUB (7)

Ventajas de la herramienta SIGIL:

- ❖ Guía del usuario en línea SIGIL y Wiki de documentación.
- ❖ Gratuito y de código abierto de software bajo licencia GPLv3.
- ❖ Multi-plataforma: se ejecuta en Windows, Linux y Mac.
- ❖ Completo soporte UTF-16.
- ❖ Completo soporte EPUB 2.
- ❖ Múltiples vistas: Vista libro, la vista Código y la vista preliminar.
- ❖ Edición WYSIWYG en el Libro View.
- ❖ Control completo sobre la edición directa sintaxis EPUB en la vista Código.
- ❖ Índice generador con varios niveles de soporte partida.

- ❖ Interfaz de usuario traducida a muchos idiomas.
- ❖ Soporta la importación de archivos EPUB y HTML, imágenes y hojas de estilo.

Book type

La herramienta Book type es una plataforma gratuita y de código abierto que permite las creaciones de eBook (en formato pdf, EPUB, Mobipocket, Odt o html). Esta herramienta permite la generación de publicaciones tanto a nivel individual como de forma colaborativa, sin límite de participantes, a través de un interfaz web muy amigable y sencilla. Otra destacable característica es que permite una fácil importación de contenido, capítulos y libros enteros de otras fuentes para reutilizar o combinar la información. En su página web, pone a disposición del usuario un manual en español y varios ejemplos. (8)

Ventajas de la herramienta Book type:

- ❖ Interfaz sencilla, limpia, de arrastrar y soltar.
- ❖ Ediciones simultáneas, con herramientas de mensajería y chat.
- ❖ Con posibilidades de exportar a Amazon, iBooks y otros servicios de impresión por demanda o libros digitales.
- ❖ Herramientas colaborativas para correctores, editores y colaboradores.
- ❖ Flujos de trabajo que mezclan impreso y digital para poder trabajar en ambas versiones.
- ❖ Control sobre el historial, versiones, copias, permisos de edición y manejo del licenciamiento.
- ❖ Posibilidad de importar contenido.

Scrivener

La herramienta Scrivener está desarrollada para Windows y MAC OS X y orientada a escritores. Esta permite concentrarse en la composición y estructura de cualquier proyecto literario o científico gracias a diferentes funciones y características. Entre ellas, permite almacenar textos de documentación y crear versiones y borradores de los propios textos. Esta herramienta dispone de una función para exportar el

documento final en diferentes formatos como Word, RTF, PDF, HTML, así como formatos de literatura digital como ePub o Kindle. (9)

Ventajas de la herramienta Scrivener:

- ❖ Introduce el texto en el orden que se desee, así como ver las partes escritas de forma individual o en su conjunto.
- ❖ Combina varios documentos individuales en un solo texto.
- ❖ Tiene un panel de corcho en el que se puede almacenar todo tipo de documentación que pueda ser de ayuda para lo que estemos escribiendo.
- ❖ Modo pantalla completa. Que permite centrarse en la escritura.
- ❖ Modo instantánea (permite ver y editar su documento y luego compararlo con versiones anteriores).

EPUB

Esta herramienta cuenta con licencia GPL y está estructurado mediante un archivo ZIP que contiene tres ficheros XML basados en estándares de código abierto, donde mediante estos se obtiene el libro electrónico generado. (10)

- ❖ Open Publication Structure (OPS): contiene las instrucciones incluidas en el propio formato del contenido.
- ❖ Open Packaging Format (OPF): describe la estructura del documento EPUB en formato XML.
- ❖ Open Container Format (OCF): almacena todos los archivos que componen un EPUB y los comprime en formato ZIP.

Ventajas de la herramienta EPUB:

- ❖ Permite representar en formato HTML el documento escogido, sin tener en cuenta las limitaciones de espacio y orientación de las diferentes aplicaciones que la representan.

- ❖ Incorpora hojas de estilo CSS para un mejor diseño de las páginas HTML.
- ❖ Permite crear un eBook con herramientas como: el Bloc de Notas y el GEDIT, al poseer formato XHTML, formato utilizado para generar páginas web.

InDesign

La herramienta InDesign de la compañía Adobe permite el diseño, edición y visualización previa de páginas en formato HTML para su impresión o distribución digital. Posee un control preciso sobre la tipografía e integra la interactividad mediante el vídeo y el audio para su reproducción en tabletas, teléfonos inteligentes y computadoras. (11)

Ventajas de la herramienta InDesign:

- ❖ Permite trabajar sobre varias hojas de edición a la vez sin que el rendimiento del programa aumente.
- ❖ Posibilita la creación de un índice sencillo por palabra clave o un índice por otro criterio más detallado de la información en el libro electrónico.

Reader EBook Wizard

La herramienta EBook Wizard guía paso a paso la creación del propio libro electrónico, a partir de ficheros en formato txt o html, con soporte también para css, jpg, bmp y gif. Brinda la opción de personalizar totalmente el libro electrónico con detalles como título, nombre de autor, categoría temática, descripción e imagen de la portada. Una vez rellenos todos los pasos del asistente, Reader EBook Wizard genera automáticamente el libro electrónico en formato LIT, compatible con Microsoft Reader. Se ejecuta sobre el sistema operativo Windows en sus versiones 95, 98, 2000 y XP. (12)

Python-Sphinx

Python-Sphinx es una herramienta generadora creada por Georg Brandl, el 21 de marzo de 2008. Es una herramienta que posee licencia BSD, lo que posibilita el uso del código fuente en software no libre y la protección del copyright en caso de posibles reclamaciones. Su principal utilización es la creación de documentación y manuales de usuario para proyectos en python. (13)

Ventajas de la herramienta Python-Sphinx:

- ❖ Genera la documentación en varios formatos como: html y texto plano.
- ❖ Permite una fácil definición de la estructura del documento, la estructura jerárquica que se puede especificar permite enlaces automáticos a los hermanos, padres e hijos o referencias a la raíz.
- ❖ Posibilita extensas referencias cruzadas, al permitir el uso de marcado semántico y vínculos automáticos de funciones, clases, citas y glosario de términos.
- ❖ Incorpora un buscador web en el mismo sitio que permite la búsqueda de palabras en el sitio web.
- ❖ Incorpora el índice general y el índice de los módulos.

A continuación se representa una tabla comparativa, que refleja las características fundamentales enfocadas en la elección de la herramienta de mayor impacto acorde a las características de la versión a implementar. Las herramientas mostradas en la propia tabla son las más utilizadas en la actualidad y son el resultado de la investigación atendiendo a las nuevas tendencias sobre la generación de libros electrónicos.

Herramientas	Sistema Operativo	Licencia	Importar Formatos: pdf o doc	Código Abierto	Inserción de niveles jerárquicos	Formatos de Salida
Jutoh	Windows+ Linux	GPL	Si	Si	No	Mobi Epub Texto plano
Sigil	Windows+ Linux	GPLv3	No	Si	No	Epub

Booktype	Windows	Corporativo	Si	No	Si	Pdf / Epub Mobi / Odt Html
Scrivener	Windows+ Linux	Propietario	No	Si	No	Doc / Rtf Pdf / Html
EPUB	Windows+ Linux	GPL	No	Si	No	Epub
InDesign	Windows	Propietario	No	No	No	Epub / Html
ReaderEbook	Windows	Freeware	No	Si	No	Epub
Python-Sphinx	Linux	BSD	Si	Si	Si	Html

Tabla 1: Características de las herramientas actuales para la generación de libros electrónicos

En la tabla que se muestra con anterioridad están reflejadas ocho herramientas que permiten la generación de libros electrónicos. Para la comparación entre las mismas se realizó una caracterización atendiendo a seis aspectos críticos para darle cumplimiento al desarrollo del sistema:

-Sistema operativo, licencia y código abierto: Con estas características se puede determinar si la herramienta a importar en la aplicación posee las mismas particularidades en cuanto a software libre o propietario, en este caso se trató por todos los medios de que la aplicación fuera multiplataforma o basada en software libre, acorde a las políticas de la Universidad.

-Importar y exportar formatos: Esta característica permite que la aplicación sea capaz de no sólo introducir los datos mediante el teclado, sino de importarlos mediante archivos. De igual manera los formatos de salida permiten la diversificación en la aplicación a la hora de exportar el proyecto en formatos independientes al propio del libro web, dándole al sistema más funcionalidad.

-Niveles jerárquicos: Esta es la característica más importante y de mayor fundamento para la

incorporación o no de estas herramientas a la aplicación, la inserción de niveles jerárquicos consiste en introducir datos organizados estructuralmente por capítulos, epígrafes y hojas o contenidos de manera automática, permitiendo esta característica contar con niveles elevados de organización en la estructura de los contenidos por parte de la aplicación.

Partiendo de esta investigación y del análisis sobre la base de los seis aspectos reflejados en la propia tabla, se define que las herramienta para darle solución al problema de la investigación de este trabajo de diploma será el motor de generador **Phyton-Sphinx** (herramienta ya definida en la primera versión y que cumple con los seis patrones de comparación), pero en una versión superior 1.1.3 a diferencia de la utilizada con anterioridad 1.0.3.

1.7. Phyton-Sphinx 1.1.3

La versión 1.1.3 del motor generador Python-Sphinx es la última producción estable y que no se encuentra en desarrollo, fue lanzada a la comunidad el 10 de marzo del 2012 con las siguientes ventajas sobre la versión 1.0.3 (12)

- ❖ Fija la función `safe_repr` para decodificar cadenas de bytes con caracteres no ASCII correctamente.
- ❖ Permite la configuración de esfinge apidoc a través de `SPHINX_APIDOC_OPTIONS`.
- ❖ Restaura la compatibilidad de Python 2.4.
- ❖ Cuando el código "python" no se especifica explícitamente, trata de analizarlo bajo ciertos patrones de comparación e interpreta fragmentos de estos códigos a Python.
- ❖ La generación no se bloquea cuando se encuentran ejemplos válidos doctest, en este caso se emitirá una advertencia.
- ❖ Reconoce y advierte sobre posibles errores circulares, en lugar de encontrarse con errores de recursividad y notificar errores inesperados
- ❖ Es capaz de restaura la compatibilidad en caso de errores inesperados.
- ❖ Fija HTMLHelp links en la entrada de índice nuevo.
- ❖ Fija ayuda mapeo idioma / codificación HTML para todos los idiomas soportados Esfinge.
- ❖ Contiene un constructor para el HTML, suprimiendo el mal comportamiento con el documento maestro en las raíces de los HTML.

- ❖ Corrige problemas de codificación de la ayuda constructor Qt.

1.8. Metodología de desarrollo

Las metodologías de desarrollo de *software* son un conjunto de procedimientos y técnicas para el desarrollo de productos de software, donde paso a paso se van ejecutando todas las actividades a realizar para lograr un producto informático deseado. El uso de una metodología permite indicar que personas deben incidir en el desarrollo de las actividades y el papel que debe tener. (14)

Entre algunas metodologías ágiles se encuentran las siguientes:

- ❖ **SCRUM:** Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en: El desarrollo de software se realiza mediante iteraciones, denominadas *Sprint*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.
- ❖ **Crystal Methodologies:** Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros).
- ❖ **Dynamic Systems Development Method (DSDM):** Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Su principal característica consiste en representar un proceso iterativo e incremental y además que el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio

del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir retroalimentación en todas las fases.

- ❖ **Adaptive Software Development (ASD):** Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: meditación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.
- ❖ **Feature-Driven Development (FDD):** Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.
- ❖ **Lean Development (LD):** Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80. Utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgosos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente.
- ❖ **OpenUp:** Es una metodología de software ágil que forma parte del Framework de modelo de proceso de Eclipse (Eclipse Process Framework), desarrollado por la fundación Eclipse, basado en el Basic Unified Process de IBM. Mantiene las características esenciales de RUP (Rational Unified Process). (15)

Después de realizar un análisis de las metodologías de desarrollo de software investigadas con anterioridad, se determina usar OpenUp por las siguientes características:

- ❖ Es una metodología ágil que tiene un enfoque centrado al cliente y con iteraciones cortas.
- ❖ Es extensible debido a que en el proceso se pueda agregar o adaptar según los requerimientos del sistema.

- ❖ Es ligero y proporciona una comprensión detallada del proyecto, beneficiando a clientes y desarrolladores sobre productos a entregar y su formalidad.
- ❖ Se centra en una arquitectura temprana para reducir al mínimo los riesgos y organizar el desarrollo.

1.9. Lenguajes de programación

La definición de lenguaje de programación consiste en un sistema de comunicación que posee una determinada estructura, contenido y uso. La programación es, en el vocabulario propio de la informática, el procedimiento de escritura del código fuente de un software. De esta manera, puede decirse que la programación le indica al programa informático qué acción tiene que llevar a cabo y cuál es el modo de concretarla. (16)

Java Script

El lenguaje de programación Java Script es un lenguaje interpretado orientado a las páginas web, con una sintaxis semejante a la del lenguaje Java. El lenguaje fue creado por Brendan Eich en la empresa Netscape Communications, empresa que fabricó los primeros navegadores de Internet comerciales. (17)

PHP

El lenguaje de programación PHP consiste en una mezcla entre interpretación y compilación para intentar ofrecer a los desarrolladores un mejor rendimiento y flexibilidad. Este lenguaje en su versión 5.3 es capaz de compilar para el código grandes cantidades de instrucciones siempre que estas sean accedidas por los propios archivos de configuración. Las principales características de PHP son: rapidez, facilidad de aprendizaje, soporte multiplataforma tanto de diversos Sistemas Operativos como de servidores HTTP y de bases de datos. PHP se distribuye de forma gratuita bajo licencia GPL. (18)

1.10. Marco de trabajo

Según Erich Gamma Marco de trabajo es “Conjunto de clases cooperativas que construyen un diseño reutilizable para un tipo específico de software”. Otra definición realizada por Ralph E. Johnson y Brian Foote explica que framework son “Aplicaciones semi-completas que pueden especializarse para producir

software a la medida. El framework describe los objetos que componen el sistema, cómo estos interactúan y cuáles son sus responsabilidades”. (19)

Symfony

El marco de trabajo Symfony es un *framework* para construir aplicaciones web con PHP. En otras palabras, Symfony es un conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones web, es una de las mejores copias para PHP del *framework* Ruby on Rails. Symfony en su versión 1.4.8 ha tomado las mejores ideas de Rails y de muchos otros framework, ha incorporado ideas propias y el resultado es un *framework* estable, productivo y posee muy buena documentación. (20)

Ext-Js

El marco de trabajo Ext-Js en su definición no es más que una librería Java Script que permite desarrollar aplicaciones complejas en Internet. Esta librería en su versión 3.3 incluye componentes UI de alto rendimiento y personalizables, modelo de componentes extensibles, un API fácil de usar y licencia código abierto (Open Source) y comerciales. (21)

1.11. Gestor de bases de datos

Un gestor de bases de datos es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, asegurando su integridad, confidencialidad y seguridad. De esta manera, manipulan de forma clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. Proveen facilidades para la manipulación de grandes volúmenes de datos. Disminuyen drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son correctamente explotados por los desarrolladores. (22)

PostgreSQL

PostgreSQL es un sistema gestor de bases de datos objeto-relacional (ORDBMS) basado en POSTGRES, desarrollado en la Universidad de California en Berkeley Departamento de Ciencias de la Computación. PostgreSQL fue pionera en muchos de los conceptos que sólo estaba disponible en algunos sistemas de bases de datos comerciales. PostgreSQL es un descendiente de código abierto del

código original de Berkeley. Es compatible con una gran parte del estándar SQL y debido a la licencia liberal, PostgreSQL 9.1 puede ser utilizado, modificado y distribuido por cualquiera de forma gratuita para cualquier propósito, ya sea privado, comercial o académico. (23)

Características:

- ❖ Disponible para Linux, UNIX y Windows 32/64bit.
- ❖ Completa documentación.
- ❖ Licencia BSD.
- ❖ Múltiples métodos de autenticación.
- ❖ Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido).
- ❖ Soporta funciones de procedimientos almacenados en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de oracle), PL/Perl, PL/Python y PL/Tcl.

1.12. Entorno de desarrollo

Un entorno de desarrollo integrado o IDE es un programa informático compuesto por un conjunto de herramientas de programación. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes y es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. (24)

NetBeans IDE

NetBeans IDE 7.2 es una aplicación de código abierto (U) diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java. NetBeans IDE dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles. (25)

1.13. Lenguaje de modelado

Partiendo del hecho que el ser humano requiere de modelos para manejar sistemas complejos, y en cuanto más complejos se vuelven los sistemas, es necesario tener mejores técnicas de modelado. El contar con una metodología universal para el desarrollo de sistemas de software es de gran beneficio en la construcción de todo tipo de sistemas. Disponer de buenos modelos facilita la comunicación entre equipos de trabajo en un gran proyecto. (26)

UML

El lenguaje de modelado unificado (UML), es una herramienta o Lenguaje de Modelamiento Unificado que permite a los creadores de sistemas generar diseños que capten sus ideas en una forma convencional y fácil de comprender y así poder comunicárselos a otra persona. (27)

1.14. Herramienta CASE

Las herramientas CASE son un conjunto de herramientas y métodos asociados que proporcionan asistencia automatizada en el proceso de desarrollo del software a lo largo de su ciclo de vida. Fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software. (28)

Visual Paradigm

Visual Paradigm for UML Enterprise Edition (VP-UML EE) 8.0 es una herramienta de desarrollo de sistemas de software todo en uno y de tecnología de la información de extremo a extremo de modelado de sistemas. Permite ayudar a los arquitectos de la empresa mediante el apoyo al modelado de procesos de negocio y las operaciones con BPMN, EPC Diagrama, captura la visión de la empresa y el alcance, la documentación de las normas y la presentación de las relaciones entre los objetos con herramientas como editor de reglas de negocio, ArchiMate, Zachman Framework, Negocios Motivación Modelo, diagrama hecho, diagrama de flujo de datos entre otros. (29)

1.15. Conclusiones parciales del capítulo

- ❖ El estudio sobre las tendencias actuales de los diferentes tipos de libros electrónicos permitió conocer que los libros web, se encuentran entre los más utilizados para visualizar información digital en navegadores web comunes como Mozilla Firefox, Internet Explorer y Chrome.
- ❖ La investigación realizada sobre las distintas herramientas utilizadas en la actualidad para la generación de los libros electrónicos, permitió establecer una caracterización partiendo de seis aspectos fundamentales, obteniendo como resultado la elección del motor Python-Sphinx en su versión 1.1.3 como motor generado, superior a la versión 1.0.3 utilizada en la herramienta anterior.
- ❖ El análisis de las tecnologías a utilizar para darle cumplimiento al objetivo general permitieron la selección de la metodología OpenUp para guiar el proceso de desarrollo del software y como herramienta para el modelado Visual Paradigm. Durante la implementación de la solución se hará uso del IDE de desarrollo NetBeans, utilizando como lenguajes de programación PHP 5.3 y Java Script soportados por los marcos de trabajos Symfony 1.4 y Ext-Js 3.3 respectivamente. Además para la definición de la base de datos se usará como gestor PostgreSQL 9.1, para la administración de la base de datos PgAdmin 1.14 y como servidor web Apache 2.2.

Capítulo 2: Análisis y Diseño

Introducción

En el presente capítulo, partiendo del análisis realizado sobre el entorno del problema de la investigación, se describen los artefactos correspondientes al flujo de trabajo análisis y diseño. Se representa el modelo del dominio, los requisitos funcionales y los no funcionales definidos para darle cumplimiento a la situación problemática. Se muestran el diagrama de caso de uso del sistema y los diagramas de clases del diseño: por medio de los cuales se muestra la estructura de la aplicación en ambos sentidos, de igual manera los diagramas de interacción modelan los aspectos dinámicos donde se representan las clases y sus relaciones.

2.1 Modelo de dominio

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales por ayudar a comprender los conceptos claves de un negocio o un dominio del problema. A continuación se presenta el modelo de dominio para la Aplicación para la Generación de Libros Web 2.0. (30). (Ver figura 1)

2.2 Descripción del modelo del dominio

- ❖ **Usuario:** Contiene los usuario que acceden a la aplicación para confeccionar sus Proyectos de Libros Web.
- ❖ **LEGA:** Aplicación para la Generación de Libros Web, en la cual se generan los manuales de ayuda web denominados Proyectos de Libros Web.
- ❖ **Proyecto Libro Web:** Prototipo del libro web que almacena los contenidos necesarios para realizar un manual de ayuda en el formato antes mencionado.
- ❖ **Motor Generador de libros web:** Encargado de obtener la información asociada ha determinado PLW y convertir la misma a un código entendible por la herramienta Python-Sphinx.

- ❖ **Libro Web:** Es el resultado final del PLW por parte de la aplicación el cual se obtiene mediante la herramienta Python-Sphinx.
- ❖ **Capítulo:** Contiene el primer nivel del proyecto del libro web creado, donde un proyecto puede tener varios capítulos.
- ❖ **Epígrafe:** Contiene el segundo nivel del proyecto del libro web creado, donde un capítulo puede tener varios epígrafes.
- ❖ **Página:** Contiene el último nivel del proyecto del libro web creado, en la cual se adicionará la información de los libros y donde un epígrafe puede tener varias páginas.

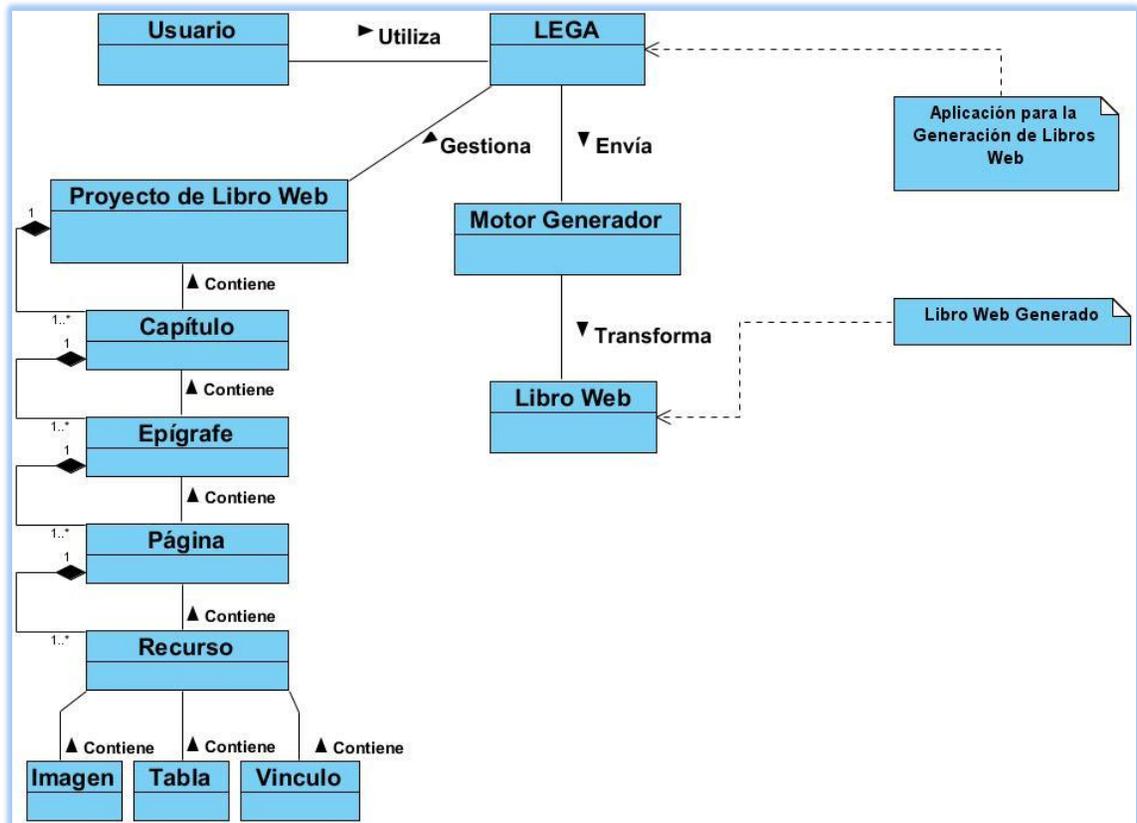


Figura 1: Modelo de Dominio

2.3 Requisitos funcionales

Los requisitos funcionales (RF) definen el comportamiento interno de un software, son condiciones que el sistema debe de cumplir. Estos muestran las funcionalidades que deben satisfacerse para cumplir con las especificaciones de software. La Aplicación para la Generación de Libros Web v2.0 debe cumplir con los requisitos funcionales que a continuación se describen. (31)

RF1. Adicionar PLW

Descripción: El sistema debe permitir la posibilidad de registrar en el sistema los datos necesarios para crear un PLW (nombre, autores, versión del proyecto, descripción y colaboradores).

Entrada: El usuario introduce los datos necesarios para crear un PLW.

Salida: El sistema almacena los datos y muestra el PLW creado.

RF2. Actualizar PLW

Descripción: El sistema debe permitir la posibilidad de registrar en el sistema los datos necesarios para modificar un PLW existente.

Entrada: El usuario introduce los datos a actualizar para determinado PLW.

Salida: El sistema almacena los datos y muestra el PLW con sus modificaciones.

RF3. Eliminar PLW

Descripción: El sistema debe permitir la posibilidad de eliminar los PLW creados.

Entrada: El usuario selecciona de la lista, el PLW que desee eliminar.

Salida: El sistema notifica si verdaderamente se quiere eliminar el PLW y se elimina en caso de ser confirmado.

RF4. Adicionar contenido.

Descripción: El sistema debe permitir la posibilidad de adicionar contenido al final de cada nivel de los PLW (capítulos, epígrafes y hojas).

Entrada: El usuario introduce los contenidos a conformar determinado PLW.

Salida: El sistema actualiza los contenidos agregados.

RF5. Actualizar contenido

Descripción: El sistema debe permitir la posibilidad de actualizar el contenido en los PLW (capítulos, epígrafes y hojas).

Entrada: El usuario actualiza los contenidos a conformar determinado PLW.

Salida: El sistema actualiza el contenido seleccionado.

RF6. Eliminar contenido

Descripción: El sistema debe permitir la posibilidad de eliminar el contenido de determinado PLW (capítulos, epígrafes y hojas).

Entrada: El usuario elimina los contenidos a conformar determinado PLW.

Salida: El sistema actualiza los contenidos agregados.

RF7. Insertar contenido

Descripción: El sistema debe permitir la posibilidad de insertar contenido en el nivel seleccionado por el usuario, en determinado PLW ya creado en el sistema.

Entrada: El usuario introducirá el tipo de contenido y su ubicación en el PLW.

Salida: El sistema actualiza los contenidos insertados.

RF8. Mostrar estadísticas de los PLW y usuarios existentes

Descripción: El sistema debe permitir la posibilidad de mostrar datos de todos los PLW creados en el sistema.

Entrada: El usuario ingresa su usuario y contraseña.

Salida: El sistema visualiza las estadísticas.

RF9. Listar los PLW

Descripción: El sistema debe permitir la posibilidad de mostrar todos los PLW.

Entrada: El usuario ingresa su usuario y contraseña.

Salida: El sistema visualiza un listado con todos los PLW.

RF10. Visualizar los PLW

Descripción: El sistema debe permitir la posibilidad de abrir los PLW creados con anterioridad.

Entrada: El usuario selecciona de la lista, el PLW que desee visualizar.

Salida: El sistema visualiza el PLW seleccionado.

RF11. Reasignar PLW

Descripción: El sistema debe permitir reasignar PLW entre usuarios.

Entrada: El usuario selecciona el usuario y el PLW a asignar.

Salida: El sistema visualiza la reasignación del PLW al usuario seleccionado.

RF12. Adicionar usuario

Descripción: El sistema debe permitir la inserción de datos para crear usuarios para el trabajo en la aplicación.

Entrada: El usuario introduce los datos pertinentes de cada usuario.

Salida: El sistema notifica la inserción correcta del usuario.

RF13. Actualizar usuarios

Descripción: El sistema debe permitir a los usuarios con permisos asignados la edición de los usuarios a los cuales tenga acceso.

Entrada: El usuario introduce los datos a modificar de determinado usuario.

Salida: El sistema notifica la correcta modificación de los datos.

RF14. Eliminar usuario

Descripción: El sistema debe permitir a los usuarios con permisos asignados eliminar usuarios a los cuales tenga acceso.

Entrada: El usuario selecciona el usuario a eliminar.

Salida: El sistema notifica si verdaderamente se quiere eliminar el usuario y se elimina en caso de ser confirmado.

RF15. Habilitar/Deshabilitar usuarios

Descripción: El sistema debe permitir a los usuarios con permisos asignados habilitar y deshabilitar usuarios a los cuales tenga acceso.

Entrada: El usuario seleccionará el usuario a habilitar o deshabilitar en el sistema.

Salida: El sistema actualizará el usuario habilitado o deshabilitado.

RF16. Exportar Proyecto Libro Web

Descripción: El sistema debe permitir exportar e importar proyectos existentes en el sistema.

Entrada: El usuario seleccionará el proyecto a exportar e importar.

Salida: El sistema actualiza el proyecto en actual ejecución.

2.4 Requisitos no funcionales

Los requisitos no funcionales (RNF) especifican criterios que pueden usarse para calificar las operaciones que un sistema realiza. Constituyen las propiedades o cualidades que el sistema debe tener (32). A continuación se muestran los requisitos no funcionales que deben cumplirse en el proceso de desarrollo del sistema.

RNF1- Usabilidad

Atendiendo a los estándares de diseño la aplicación debe poseer una interfaz agradable con alto grado de usabilidad, permitiendo una mejor comprensión de las funcionalidades que brinda el sistema. Debe ser sencillo y fácil de usar.

RNF2- Software

Programas necesarios para el servidor web:

- ❖ Sistema Operativo: Ubuntu GNU/Linux 8.04 o superior (Plataforma Libre).
- ❖ Paquetes: apache2, php5, libapache2-mod-php5, php5-cli, php5-mysql, php5-pgsql, php5-sqlite, php5-sybase, php5-xsl, php5-gd, php-apc, Python-Sphinx.
- ❖ Navegador: Mozilla Firefox versión 4.0 o superior.

Programas necesarios para el servidor de bases de datos:

- ❖ Sistema Operativo: Ubuntu GNU/Linux 8.04 o superior (Plataforma Libre).
- ❖ PostgreSQL versión 9.0 o superior.
- ❖ PGAdmin III o algún administrador para PostgreSQL
- ❖ PostgreSQL debe estar configurado para aceptar conexiones vía TCP/IP utilizando el método de autenticación por md5.

Programas necesarios para los ordenadores clientes:

- ❖ Sistema Operativo: Ubuntu GNU/Linux 8.04 o superior (Plataforma Libre). Sistema Operativo Windows de Microsoft en su versión XP o superior.
- ❖ Los clientes tendrán acceso al sistema a través de los navegadores web: Internet-Explorer, Mozilla Firefox, Opera u otro compatible, preferentemente el Mozilla Firefox superior a la versión 8 y la comunicación de las terminales clientes con el servidor será a través de conexiones de redes UTP a una velocidad constante de 100 Mb/s o superior.

RNF3- Hardware

Requerimientos mínimos de hardware para los servidores:

- ❖ El servidor web necesita 1GB de RAM y un procesador de 2.0 GHZ, de igual manera el servidor de BD requiere de los mismos recursos y de capacidad libre en disco duro. (10 GB para instalar el sistema)

Requerimientos mínimos de hardware para los clientes:

- ❖ Los ordenadores clientes necesitan 512 MB de RAM y un procesador Pentium 4 a 1.7 GHz.

RNF4- Restricciones de diseño e implementación

Se establece como marco de trabajo para el desarrollo del sistema del lado del servidor lo siguiente:

- ❖ La aplicación deberá ser implementada en el lenguaje PHP 5.3. Como «framework» de desarrollo se utilizará Symfony 1.4 el cual propone una arquitectura modelo, vista, controlador (MVC).

Se establece como marco de trabajo para el desarrollo del sistema del lado del cliente lo siguiente:

- ❖ La aplicación deberá ser implementada con el «framework» para el desarrollo web Ext-Js 3.3, permitiendo el diseño de interfaces visuales con apariencia web de escritorio.

RNF5-Apariencia o interfaz externa

Se desea una aplicación web con interfaz amigable, fácil de usar y acorde a las políticas del departamento.

Ver anexo 3 y 4 para visualizar interfaces de la Aplicación para la Generación de Libros Web 2.0.

RNF6-Disponibilidad

El sistema debe permanecer en funcionamiento en los horarios de trabajo establecidos por las entidades, excepto cuando sea necesario reiniciarlo o detenerlo por tareas de mantenimiento o cambio de configuración. Algunos errores que pueden resultar críticos son:

- ❖ Que se encuentre fuera de servicio el servidor que almacena la base de datos con la que trabaja el sistema o que no exista conectividad hacia la BD.

- ❖ Que se encuentre fuera de servicio el servidor de aplicaciones.

RNF7-Seguridad

La información manejada por el sistema debe estar protegida de acceso no autorizado y de igual manera los datos tienen que estar disponible sólo para los usuarios que pertenezcan al sistema y estén registrados en la base de datos. A través de la autenticación correcta se podrá obtener la información solicitada.

RNF8- Rendimiento

La aplicación debe ser rápida y el tiempo de respuesta debe afectar lo menos posible el trabajo del usuario.

Tiempo máximo de respuesta para la generación de un libro web:

El tiempo máximo para la generación de un libro web no debe sobrepasar un minuto.

Tiempo promedio de respuesta para la generación de un libro web:

El tiempo promedio de respuesta oscila en un intervalo de 3-10 segundos en dependencia del tamaño del Manual de Ayuda que se desee generar.

Cantidad de usuarios conectados de forma simultánea:

El sistema debe permitir que existan al menos 25 usuarios conectados de forma simultánea.

RNF9- Instalación y capacitación

Se garantiza la instalación del sistema y se impartirá capacitaciones a los usuarios interesados. Además, se cuenta con un manual de usuario y de instalación que guían paso a paso las acciones de los usuarios. También, se le dará seguimiento al funcionamiento de la aplicación durante un período de tiempo determinado.

RNF10- Requisitos legales y derecho de autor

Al finalizar el desarrollo de la Aplicación y se haya identificado el proyecto que pueda utilizar la solución, la Dirección del Centro DATEC definirá las normas del contrato por la que se registrará el mismo.

2.5 Diagrama de caso de usos del sistema

El diagrama de casos de uso del sistema (DCUS) se utiliza para describir las funcionalidades de un software y documentar su comportamiento. Los casos de uso (CU) engloban los requisitos funcionales de un sistema, representando las funciones que la aplicación puede realizar. (Ver figura 2)

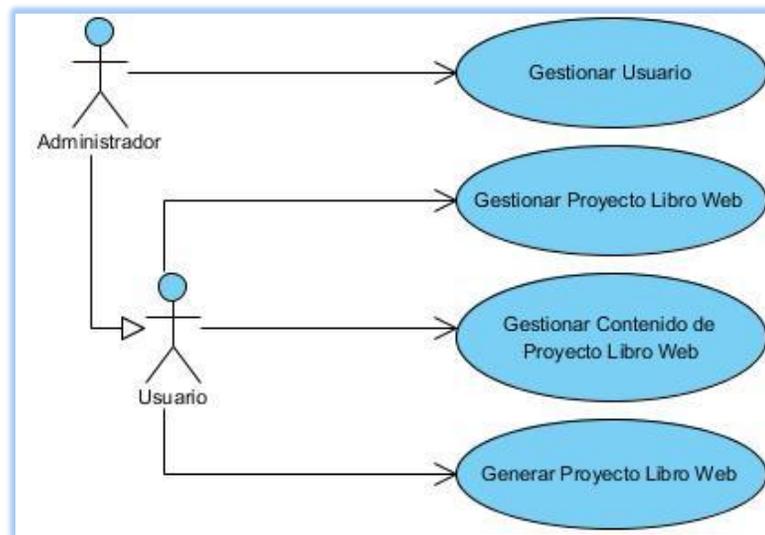


Figura 2: Diagrama de Caso de Uso del Sistema (DCUS).

En este DCUS, el actor Usuario es el encargado de inicializar entre otros casos de usos el de Gestionar Contenido de proyecto Libro Web y Generar Proyecto Libro Web, considerados casos de usos críticos por su impacto en la arquitectura del sistema.

A continuación se describirá el caso de uso:

Gestionar Contenido de proyecto Libro Web: Permite adicionar, insertar, modificar y eliminar el contenido de su estructura en determinado PLW existente.

Caso de Uso:	Gestionar contenido de PLW
---------------------	----------------------------

Actores:	Usuario , Administrador	
Resumen:	El presente caso de uso inicia cuando el actor comienza a incorporarle contenido a determinado PLW.	
Precondiciones:	En el sistema tienen que existir un PLW a los cuales se le insertarán contenidos.	
Referencias	RF-4, RF-5, RF-6	
Prioridad	Critico	
Flujo Normal de Eventos		
Sección 1“Adicionar contenido”		
Acción del Actor	Respuesta del Sistema	
1. El actor accede a adicionar un nuevo contenido en determinado PLW.	2. El sistema muestra todos los posibles tipos de contenidos a adicionar.	
3. El actor selecciona determinado contenido a incorporar al PLW y le introduce un nombre.	4. El sistema visualiza en el PLW seleccionado el nuevo contenido insertado y notifica la correcta inserción en caso de que se hayan validado correctamente los datos.	
Prototipo de Interfaz		



Figura 3: Prototipo de Interfaz: Escenario Adicionar Contenido

Flujos Alternos (paso 4: " Error en la inserción de datos, no pueden insertar caracteres especiales)

Acción del Actor	Respuesta del Sistema
	4.1 El sistema notifica que existieron errores en la entrada de datos.
5. El actor introduce nuevamente los datos.	6. El sistema valida los nuevos datos introducidos y notifica la correcta inserción.

Prototipo de Interfaz

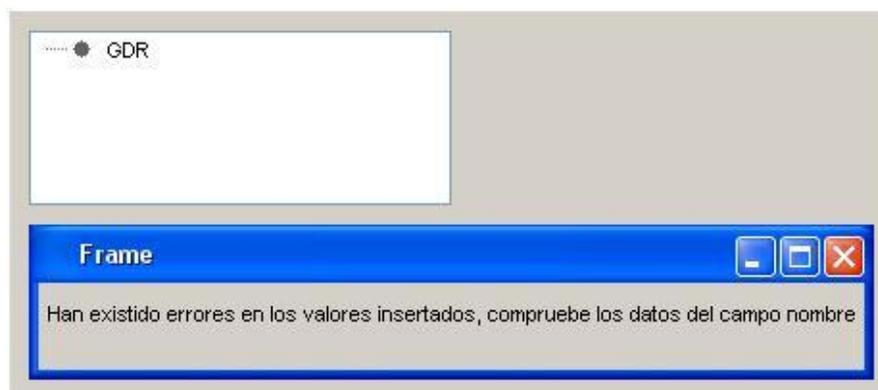


Figura 4: Prototipo de Interfaz: Escenario Adicionar Contenido (Flujo Alternativo)

Sección 2 “Actualiza contenido”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona un contenido en determinado PLW.	2. El sistema permite cambiar el nombre de este contenido.
3. El actor introduce los nuevos datos.	4. El sistema valida dichos datos y notifica la correcta modificación del contenido.

Prototipo de Interfaz



Figura 5: Prototipo de Interfaz: Escenario Actualizar Contenido

Flujos Alternos (paso 4: “ El contenido se ha modificado de manera incorrecta, no pueden insertarse caracteres especiales)

Acción del Actor	Respuesta del Sistema
	4.1 El sistema notifica el error en la validación.

5 El actor nuevamente inserta los datos a modificar.	6. El sistema valida los datos y notifica la correcta modificación o no de estos.
--	---

Prototipo de Interfaz

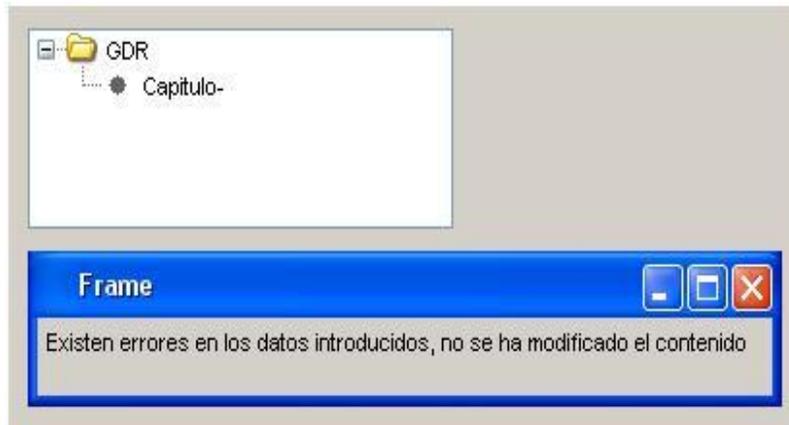


Figura 6: Prototipo de Interfaz: Escenario Actualizar Contenido (Flujo Alternativo)

Sección 3 “Eliminar contenido”

Acción del Actor	Respuesta del Sistema
1. El actor selecciona un contenido en determinado PLW.	2. El sistema permite eliminar el contenido seleccionado.
3. El usuario elimina el contenido	4. El sistema visualiza el PLW sin el contenido eliminado.

Prototipo de Interfaz

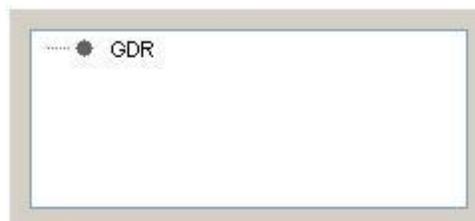


Figura 7: Prototipo de Interfaz: Escenario Eliminar Contenido

Sección 4 "Insertar contenido"	
Acción del Actor	Respuesta del Sistema
1. El actor notifica la inserción de un contenido antes o después del determinado contenido.	2. El sistema muestra todos los posibles tipos de contenidos a agregar y la posición donde se quiere insertar.
3. El actor selecciona determinado contenido a incorporar al PLW y le introduce su nombre, al igual que su ubicación.	4. El sistema visualiza en el PLW seleccionado el nuevo contenido insertado y notifica la correcta inserción en caso de que se haya validado correctamente los datos.

Prototipo de Interfaz



Figura 8: Prototipo de Interfaz: Escenario Insertar Contenido

Flujos Alternos (paso 4: " El contenido se ha insertado de manera incorrecta, no puede contener caracteres especiales)

Acción del Actor	Respuesta del Sistema
	4.1 El sistema notifica que existieron

	errores en la entrada de datos.
5. El actor introduce nuevamente los datos.	6. El sistema valida los nuevos datos introducidos y notifica la correcta inserción.
<p><i>Prototipo de Interfaz</i></p> 	
<p>Figura 9: Prototipo de Interfaz: Escenario Insertar Contenido (Flujo Alternativo)</p>	
Poscondiciones	PLW almacenado con nuevo contenido o sin parte de este.

Tabla 2: Descripción del caso de uso Gestionar Contenido de PLW

2.6 Patrones utilizados en la aplicación

A continuación se describe el patrón arquitectónico, los patrones de casos de usos y los patrones de diseño que mejor se acoplan a las características y restricciones del sistema.

Patrones de casos de uso:

CRUD (completo): Este patrón es utilizado cuando las operaciones pueden ser realizadas sobre una parte de la información de un tipo específico o sobre toda la referente a ésta, un ejemplo del uso de este patrón es en el caso de uso Gestionar usuario, donde se agrupan los requisitos funcionales de crear, modificar, leer y eliminar usuarios.

Múltiples actores (rol común): Este patrón es utilizado cuando más de un actor convergen en un mismo rol sobre un caso de uso determinado, un ejemplo de esto es el usuario y administrador, quienes pueden acceder de igual forma los casos de uso como generar PLW siendo actores independientes.

Patrones de arquitectura

Modelo Vista Controlador (MVC): El patrón separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, ventaja que posibilita el mantenimiento de los sistemas, la seguridad y la simplificación en el desarrollo.

En el presente trabajo de diploma se utiliza el patrón MVC, estructurado de la siguiente manera, en la capa **VISTA** estarán todos los componentes .js propios del subsistema Ext-Js, donde toda la interacción de los usuarios se realizará mediante estos. En la capa **MODELO** se incluirán todos los datos con los que trabaja el sistema, que no son más que las informaciones de los distintos objetos almacenados en la base de datos y por último, en la capa **CONTROLADOR** estarán todas las clases controladoras que reciben las peticiones de los usuarios y son respondidas luego de comunicarse con el MODELO en caso de ser necesario.

Patrones de diseño:

GRASP: (Patrones de Software para Asignar Responsabilidades) son aquellos que indican y describen los principios de diseño de objetos para la asignación de responsabilidades.

Los Patrones GRASP a utilizar en la aplicación son:

Controlador: Este patrón actúa como intermediario entre una interfaz específica y el algoritmo que la implementa, recibiendo los datos del usuario y enviándolos a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, para aumentar la reutilización de código. Con este patrón cada clase posee responsabilidades específicas de controlar el flujo de eventos del sistema. Symfony aplica este patrón permitiendo una estructura bien organizada desde el index.php del ambiente hasta las actions. Cada clase en esta capa tiene su responsabilidad y es única. En la aplicación se utiliza este patrón en las clases que tienen responsabilidades específicas de controlar el flujo de eventos del sistema.

Bajo Acoplamiento: Consiste en que las dependencias entre las clases deben ser las menos posibles, de manera que en caso de producirse una modificación en alguna de las mismas, se tenga la mínima repercusión posible en el resto de estas. Con un bajo acoplamiento las clases de acceso a datos son independientes de las de abstracción a datos, es decir que si se modifican las clases del modelo no se afectan las clases del controlador. Este patrón se evidencia en la capa modelo al existir una independencia entre las clases de acceso a datos y las de abstracción a datos, posibilitando una mayor reutilización.

Alta Cohesión: Consiste en que una clase tiene una única labor dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. La información que almacena una clase debe ser coherente y estar en mayor medida relacionada con la clase. Estas clases mejoran la claridad y la facilidad con que se entiende el diseño. Este patrón se evidencia en la clase gestionar usuario, la cual tiene la responsabilidad de contener todas las acciones asociadas a los datos de los usuarios, independiente de los proyectos que tenga creado.

Experto: Consiste en que una clase tiene que tener una responsabilidad determinada, ni muchas ni pocas, simplemente la que le corresponde hacer. Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Este patrón se encuentra presente en la capa modelo donde existen dos tipos de clases las de abstracción de datos y las de acceso a datos que trabajan directamente con la base de datos. El *framework* Symfony usa el ORM propel para el mapeo de las tablas de la base de datos, creándose cuatro clases. Las de acceso a datos que trabajan directamente con la base de datos y las de abstracción de datos que son las que poseen los atributos necesarios para realizar dicha función. Por ejemplo la clase BaseUsuario no conoce los atributos para interactuar con la base de datos tan solo se debe implementar la responsabilidad en la que se le avise a la clase BaseUsuarioPeer que es la contenedora de los datos necesarios para ejecutar la acción.

Patrones de diseño:

GOF (Banda de los Cuatro) dan una descripción de clases y objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular.

Los Patrones GOF utilizados en la aplicación son:

Decorator: Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. Es muy importante debido a que está presente en el layout de la aplicación, el cual contiene el código HTML y decora según sea la petición las demás plantillas. Este patrón está presente en el layout el cual contiene el código HTML incluido y va decorando todas las plantillas a utilizar de acuerdo con la petición del usuario.

Front Controller: Es uno de los patrones de diseño más importantes debido a que constituye el único punto de acceso a la aplicación. Es el encargado de recibir las peticiones, controlar y gestionar las peticiones web realizadas por los clientes. Este patrón se visualiza al utilizar el *framework* Symfony, constituyendo este el único punto de acceso a la aplicación. Además, recibe las peticiones y decide el flujo a seguir.

2.7 Diagrama de clases del diseño

Los diagramas de clases del diseño son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. Describen las especificaciones de las clases del software y de las interfaces en una aplicación. (29). (Ver figura 10)

Ver anexo 2 para visualizar otros diagramas de clases del diseño.

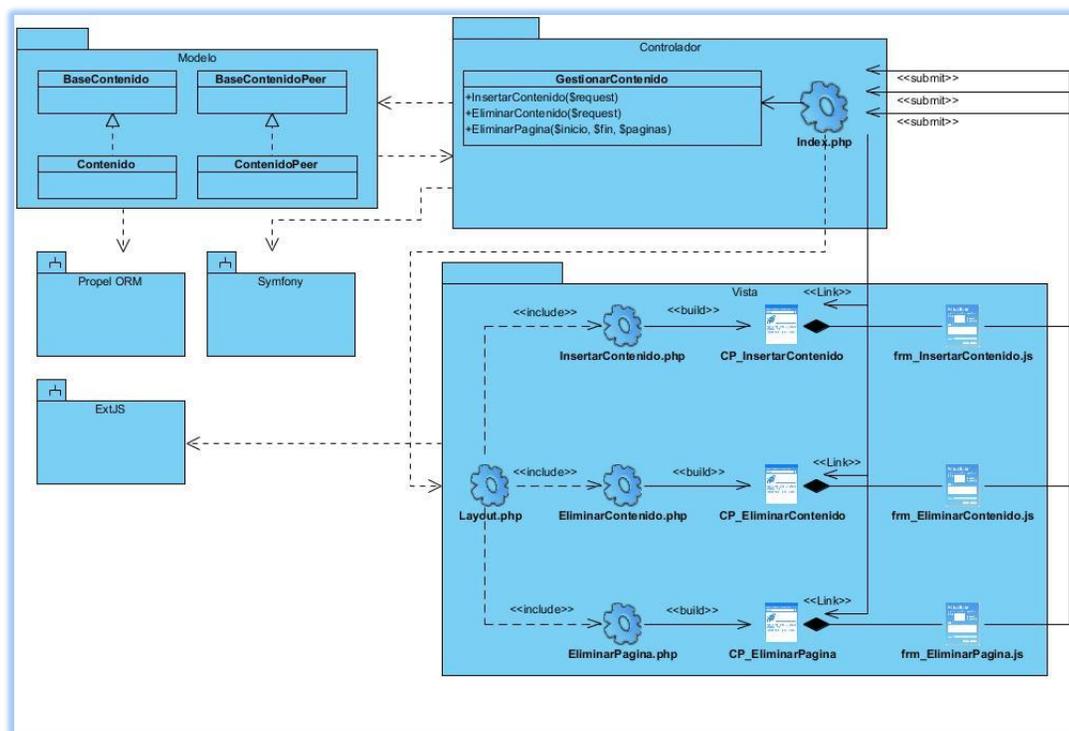


Figura 10: Diagrama de clase del diseño del CU: Gestionar Contenido.

En este diagrama se muestran las clases agrupadas atendiendo al Modelo Vista Controlador, la clase Modelo es la que contiene todas las clases de la base de datos. La clase Controladora es la que recibe las peticiones de solicitudes al modelo y realiza las acciones necesarias para obtener los datos del modelo para enviarlos a la vista, que es la encargada de mostrar la información al usuario. La clase vista es la que almacena todas las interfaces de usuario, las cuales se encuentran distribuidas en componentes. Las solicitudes se realizan por medio de la tecnología AJAX (Java Script asíncrono y XML) y las respuestas son devueltas específicamente en formato JSON (Java Script Objeto Notación). Los elementos del lado del cliente corresponden a componentes específicos del negocio como paneles de trabajos asociados al caso de uso y a los componentes genéricos fuertemente reutilizables e implementados en lenguaje Java Script utilizando los componentes Ext-Js.

2.8 Diagrama de secuencia

Los diagramas de secuencias muestran la forma en que un grupo de objetos interactúan entre sí a lo largo del tiempo. Un diagrama de secuencia consta de objetos, mensajes entre estos objetos y una línea de vida del objeto representada por una línea vertical. (33). (Ver figura 11)

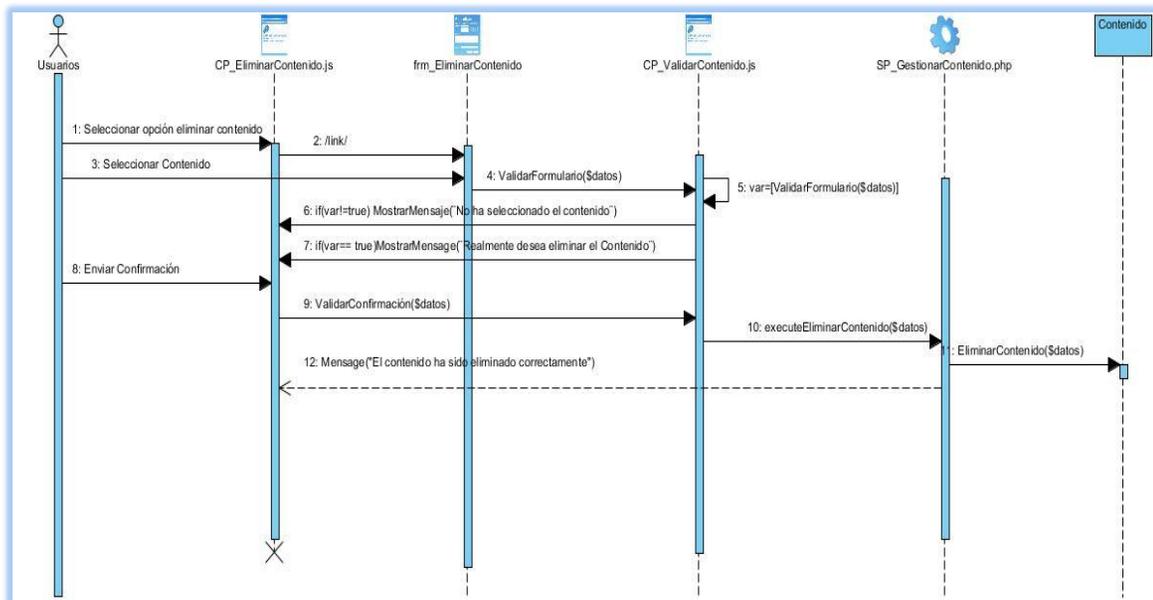


Figura 11: Diagrama de secuencia del escenario eliminar contenido.

En este diagrama el actor involucrado, para este caso el usuario, puede crear, modificar o eliminar el contenido de determinado libro web, en este escenario el actor se dispone a eliminar un proyecto creado en el sistema, solicitando a la página cliente eliminar un específico libro web y este mediante la controladora devuelve a la interfaz el mensaje de confirmación de la eliminación de determinado proyecto, donde luego de la respuesta afirmativa se elimina el proyecto seleccionado con anterioridad.

2.9 Modelo de datos

Un modelo de datos es un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos. Es formal pues los objetos del sistema se manipulan siguiendo reglas perfectamente definidas y utilizando exclusivamente los operadores definidos en el sistema, independientemente de lo que estos objetos y operadores puedan significar. (34) A continuación se

muestra el modelo de datos del presente trabajo de diploma. (Ver figura 12)



Figura 12: Modelo de datos

A continuación se realizara una breve descripción de todos los campos de cada tabla, reflejada en el modelo de datos.

Descripción de la Tabla Trazas	
Idtraza	Representa la llave primaria de cada traza.
Actividad	Representa la actividad realizada por cada traza de la tabla.
Fecha	Representa la fecha de ejecución de cada traza de la tabla.
Horario	Representa el horario de ejecución de cada traza de la tabla.
Usuarioidusuario	Representa el identificador del usuario propietario de la traza.
Ip	Representa el Ip de donde se ejecutó cada traza de la tabla.

Tabla 3: Descripción de la Tabla Traza del modelo de datos

Descripción de la Tabla Proyectos	
Idproyecto	Representa la llave primaria de cada proyecto de la tabla.
Nombrepryecto	Representa el nombre del proyecto de cada proyecto de la tabla.
Autor	Representa el nombre del autor de cada proyecto de la tabla.
Versión	Representa la versión de cada proyecto de la tabla.
Colaboradores	Representa los colaboradores (no autores) de cada proyecto de la tabla.
Descripción	Contiene una descripción de que trata cada proyecto de la tabla.
Plantilla	Contiene la plantilla predeterminada seleccionada por el usuario.
XML	Contiene toda la estructura del proyecto (Capítulos, epígrafes, hojas)
Abierto	Representa un booleano para identificar si está abierto por el usuario.
Usuarioidusuario	Representa la llave primaria del usuario que contiene el proyecto.
Fcreacion	Representa la fecha de creación de cada proyecto de la tabla.
Fmodificacion	Representa la fecha de la última modificación de cada proyecto.
Generado	Representa si el proyecto ha sido generado en al menos una ocasión.
Ip	Representa el IP de donde se introdujo cada proyecto de la tabla
Hora	Representa la hora en que fue registrado cada proyecto de la tabla.
Control	Representa un booleano para identificar si se está ejecutando.
Ipactual	Representa el IP real host donde está el usuario actualmente.

Tabla 4 Descripción de la Tabla Proyectos del modelo de datos

Descripción de la Tabla Usuarios	
Idusuario	Representa la llave primaria de cada usuario.
Usuario	Representa el nombre del usuario de cada usuario de la tabla.
Nombre	Representa el nombre de cada usuario de la tabla.
Apellidos	Representa los apellidos de cada usuario de la tabla.
Contraseña	Representa la contraseña de cada usuario de la tabla.
Privilegio	Representa el privilegio administrativo de cada usuario de la tabla.
Modoldap	Representa si el usuario está registrado bajo el modo Ldap o no.
Centro	Representa el centro de producción de cada usuario de la tabla.
Habilitado	Representa si el usuario está habilitado en el sistema
Admin	Representa si el usuario tiene permisos de jefe de centro.
Departamento	Representa el departamento a que pertenece cada usuario.

Tabla 5 Descripción de la Tabla Usuarios del modelo de datos

2.10 Diagrama de despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. (35) (Ver figura 13)

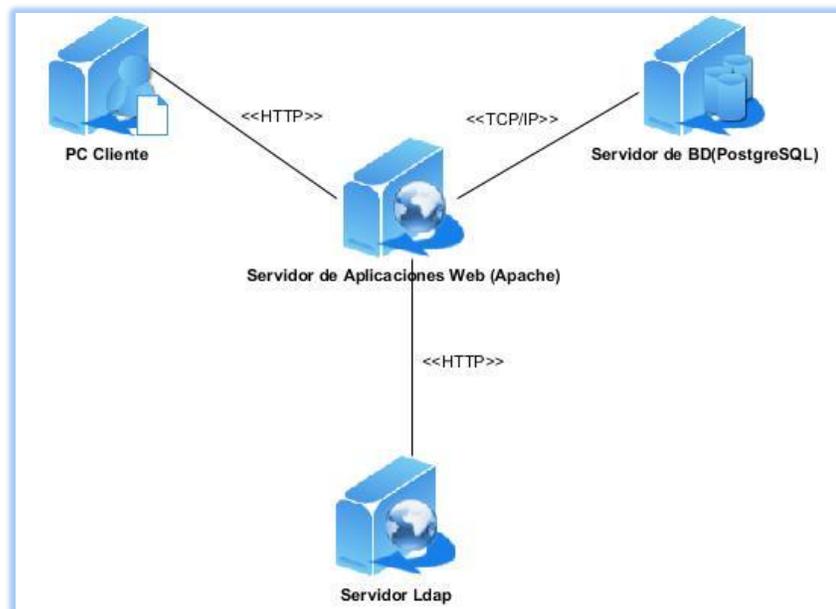


Figura 13: Diagrama de despliegue

En este caso se necesitan dos servidores: uno web con entorno de ejecución Apache2, con soporte para PHP 5.3 y sistema operativo basado en GNU/Linux (en cualquiera de sus distribuciones) y el otro servidor para contener la base de datos, la cual necesita el gestor de BD PostgreSQL, versión 8.4 o superior. Los usuarios podrán conectarse a la aplicación desde estaciones clientes mediante el protocolo HTTP, con igual protocolo se comunican los servidores LDAP y de aplicaciones, mientras que por el TCP/IP, se accede al de bases de datos.

2.11 Conclusiones parciales del capítulo.

- ❖ En el desarrollo del presente capítulo se realizó una representación visual de clases conceptuales del entorno real de los objetos del sistema a través del modelo de dominio. De igual manera se describieron los requisitos funcionales y no funcionales que el sistema debe cumplir.
- ❖ Para comprender la estructura del sistema se realizó el diagrama de caso de uso del sistema, donde se agruparon 16 requisitos funcionales en cuatro casos de usos, considerados dos de ellos críticos para la aplicación. En este capítulo fueron realizados los diagramas de secuencia que son empleados en la visualización de los aspectos dinámicos de la aplicación en sentido general.

- ❖ En el presente capítulo se identificaron los patrones de casos de uso aplicados en el propio diagrama, y de igual manera los de diseño y de arquitectura a utilizarse en el desarrollo del trabajo, se modeló la vista de despliegue teniendo en cuenta que el sistema utiliza tres nodos principales, un servidor de aplicaciones, un servidor LDAP y el ordenador del cliente.

Capítulo 3: Implementación y Pruebas

Introducción

El presente capítulo tiene como tarea desarrollar los artefactos correspondientes a la implementación, tomando como entrada los resultados obtenidos en la etapa de diseño. Se realizará una descripción de cómo los elementos del modelo de diseño se implementan en términos de componentes, así como los resultados de los correspondientes casos de pruebas aplicados al sistema

3.1 Modelo de implementación

El modelo de implementación no es más que un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente, entre otros. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

3.2 Diagrama de componentes

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos y muestra las dependencias entre estos. Cada subsistema corresponde a un paquete físico y cada componente a un módulo, fichero o librería existente en la memoria de almacenado. (30)

Seguidamente se expone el diagrama de componente del CU “Gestionar contenido” el cual cuenta con 3 paquetes de implementación básicos. Estos paquetes son:

- ❖ El Paquete de Clases Vista, que agrupa los componentes que permiten la interacción directa con el usuario final del sistema, mostrando y recogiendo información.
- ❖ El Paquete de Clases Controlador, que contendrá la clase que manipula los eventos del usuario y realiza peticiones al subsistema modelo para mostrarlas en las vistas.
- ❖ El Paquete de Clases Modelo, que agrupa las clases que interactúan con la base de datos y velan por el cumplimiento de las reglas del negocio. (Ver figura 14)

Ver anexo 1 para visualizar diagramas de componentes.

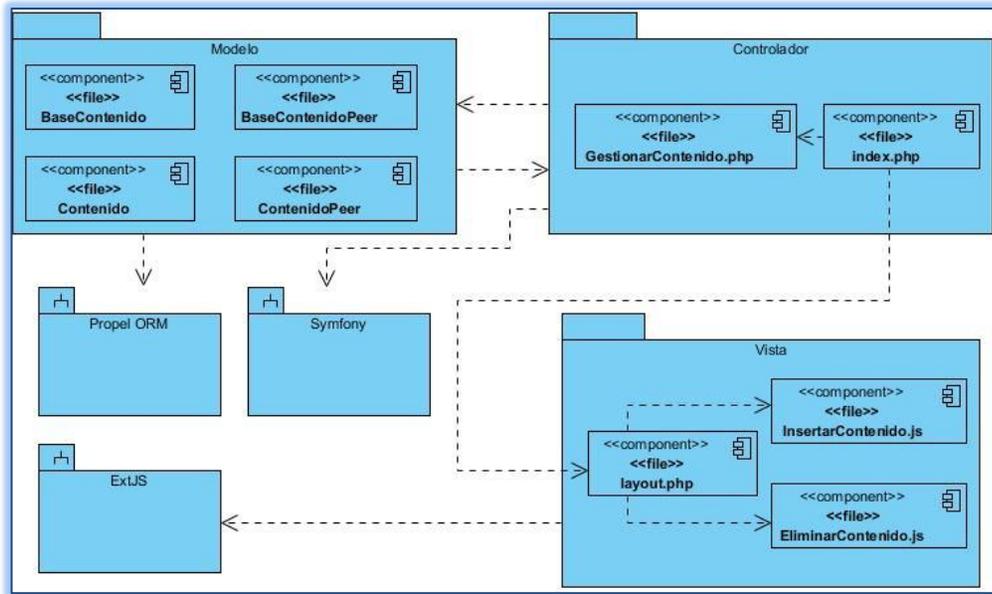


Figura 14: Diagrama de componentes del CU: Gestionar Contenido

3.3 Estándares de codificación

Un estándar de codificación son reglas que se siguen para la escritura del código fuente y comprende todos los aspectos de la generación de código. Es importante establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada y todo se realice bajo un mismo estándar.

Luego de aplicar el estilo de codificación definido, se logró que el código fuente cuente con las siguientes cualidades:

3.3.1 Notación CamelCase

La notación CamelCase consiste en escribir los identificadores con la primera letra de cada palabra en mayúsculas y el resto en minúscula. (Ver figura 15)

3.3.2 Espacios en blancos y declaraciones por línea

Se debe usar una línea en blanco entre los métodos, variables locales de un método y la primera sentencia y por último debe existir de igual manera un espacio entre las diferentes secciones lógicas dentro de un fichero, permitiendo mayor legibilidad (ver figura 16). De igual manera cada variable será declarada en una línea distinta, de esta forma cada variable se puede comentar por separado. (Ver figura 17)

3.3.3 Ejemplos de códigos fuente.

```
if ($dominio->getUser() != "") {
    $id = $this->getUltimoIdUsuario() + 1;
    $us = new Usuario();
    $us->setIdusuario("USR" . $id);
    $us->setNombre($dominio->getNombre());
    $us->setApellidos($dominio->getApellidos());
    $us->setUsuario($usuario);
    $us->setPrivilegio($privilegio);
    $us->setModoldap(true);
    $us->save();
    $this->CrearAreaTrabajo($usuario);
    return $this->renderText(json_encode(array(
        'success' => true,
        'msg' => 'Usuario insertado satisfactoriamente',
    )));
}
```

Figura 15: Fragmento de código con la notación CamelCase.

```
public function executeGetUsuariosP() {...}

public function executeObtenerusuario(sfWebRequest $request) {...}

public function executeObtenertrazas(sfWebRequest $request) {...}
```

Figura 16: Fragmento de código con espacios en blancos.

```
$datosProyecto = json_decode($request->getParameter('proyecto'));
$nombre = $datosProyecto->nombre;
$autor = $datosProyecto->autor;
$version = $datosProyecto->version;
$colaboradores = $datosProyecto->colaboradores;
$descripcion = $datosProyecto->descripcion;
```

Figura 17: Fragmento de código con declaraciones por líneas

```
if ($tipoContenido == "capitulo") {
    $padrecapitulo = $especifico->parentNode;
    $capitulos = $padrecapitulo->getElementsByTagName('capitulo');
    $cantidadcapitulos = $capitulos->length;
    for ($i = $pos; $i < $cantidadcapitulos; $i++) {
        $capituloactual = $capitulos->item($i);
        $nombrecap = $capituloactual->getAttribute('nombrecapitulo');
        $nombrecap = explode("-", $nombrecap);
        $nombrecap = $nombrecap[0] . "-" . $i;
        $capituloactual->setAttribute('numero', $i);
        $capituloactual->setAttribute('nombrecapitulo', $nombrecap);
        $capituloactual->setAttribute('id', 'CPT' . ($i));

        $epigrafas = $capituloactual->getElementsByTagName('epigrafe');
        $cantepigrafas = $epigrafas->length;
```

Figura 18: Fragmento de código del CU: Gestionar Contenido

3.4 Pruebas de software

El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores. Además son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa. (36)

3.4.1 Niveles de prueba

Los niveles de pruebas verifican y validan un producto de software en diferentes ángulos, tienen la capacidad de determinar no conformidades en relación con el uso que se le vaya a dar a la herramienta, así como del cumplimiento de todas las descripciones y respuestas del sistema en diferentes entornos.

Entre los diferentes niveles de prueba se encuentra los siguientes. (37)

- ❖ Prueba Aceptación
- ❖ Prueba de Integración
- ❖ Prueba de Unidad
- ❖ Prueba Sistema
- ❖ Prueba de Desarrollador

3.4.2 Tipos de prueba

Cada nivel de prueba engloba una técnica o tipo de prueba específica. Entre los diferentes tipos de pruebas se encuentran los siguientes: (37)

- ❖ Funcionalidad
- ❖ Rendimiento (Carga y Stress)
- ❖ Usabilidad
- ❖ Soporte
- ❖ Fiabilidad

3.4.3 Método de prueba

Existen métodos de prueba independientemente del tipo que se utilice o el nivel en que se enmarquen estas técnicas. Estos métodos proporcionan distintos criterios para generar casos de prueba que propicien fallos en los programas, agrupándose en: (37)

- ❖ Método de Caja Blanca o Estructural: Se basa en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.
- ❖ Método de Caja Negra o Funcional: Se realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.

Para realizar las pruebas al presente trabajo de diploma fue seleccionado el nivel de desarrollador, haciendo uso de los tipos o técnicas de prueba siguientes:

Funcionalidad: encargada de probar una funcionalidad completa donde pueden estar implicadas una o varias clases y la propia interfaz de usuario.

Seguridad: reconocerán si los roles definidos en el sistema, al igual que los permisos, se corresponden entre ellos.

Rendimiento: son usadas para validar y valorar la aceptación de los límites operacionales de un sistema bajo carga de trabajo.

Método de Caja Negra son las pruebas que se llevan a cabo sobre la interfaz del software. Las mismas se centran en los requisitos funcionales y permiten obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales.

3.4.4 Resultados de las pruebas

Pruebas de rendimiento (Carga y Stress)

Para realizar la prueba de rendimiento se utilizó la herramienta JMeter, la cual es capaz de mediante la inserción de un número de usuarios determinado, realizar un informe del tiempo de demora promedio para funcionalidades seleccionadas en el sistema, en este caso particular se realizaron pruebas a todas las funcionalidades, propiciando como resultado un tiempo de espera de 2.1 segundo, tiempo considerable y asequible atendiendo a los tiempos de espera propuestos.

En la siguiente tabla se reflejan dos funcionalidades de las muestras totales probadas, en este caso son el requisito Adicionar Proyecto y Adicionar Usuario, ambos con peticiones simultáneas de 50 usuarios y con los tiempos de respuesta mostrados.

Sesiones	Muestras	Solicitudes	Tiempo de respuesta (m/s)
Insertar Proyecto	50	50	872
Insertar Usuario	50	50	988

Tabla 6: Resultados de las pruebas de rendimiento con la herramienta JMeter.

Pruebas de seguridad

Para la prueba de seguridad se emplearon usuarios con diferentes roles, con el objetivo de visualizar si era permisible el acceso a funciones no correspondientes a los permisos definidos para cada usuarios, luego de varios intentos los resultados fueron los esperados, donde el acceso no fue aceptado en las variantes probadas y descritas a continuación.

Casos de prueba	
Acción	Respuesta
Se accede al sistema con un usuario sin privilegio, tratando de acceder a funciones que necesiten dicho privilegio, como eliminar usuario, reasignar proyecto y habilitar y deshabilitar usuario.	Esta acción se ejecutó varias veces, donde cada rol solo puede ejecutar sus funciones de acuerdo a lo previsto.
Se accede al sistema con usuarios válidos pero sin sus contraseñas correctas, con usuarios no válidos y contraseñas correctas tanto en el modo local como en el modo LDAP.	Esta acción no permitió registrar en el sistema los usuarios con los que se realizaron las pruebas.

Tabla 7: Resultados de las pruebas de seguridad

Método de caja negra

Luego de realizar el método de caja negra basados en casos de prueba y utilizando la técnica de partición de equivalencia, se comprobó el correcto funcionamiento de los casos de uso con sus respectivos escenarios. Fueron encontradas 3 no conformidades reflejadas en la siguiente tabla.

Casos de Prueba	Iteraciones	No. NC	Estado de NC
Gestionar Usuario	2	2	Resuelta
Gestionar Proyecto	2	1	Resuelta

Tabla 8: Resultados de las pruebas de Caja Negra

3.5 Conclusiones parciales del capítulo

- ❖ El desarrollo del presente capítulo permitió realizar la descripción de la implementación del módulo, representando las dependencias que existen entre los principales componentes, a través del diagrama de componentes correspondiente a cada caso de uso.
- ❖ Se detalló el uso de los estándares de codificación, el cual es de gran importancia para la calidad del software y para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada, la notación CamelCase (notación camello) permite homogenizar toda la estructura del código en el sistema, al igual que las aclaraciones por líneas y espacios en blancos definidos.
- ❖ Se obtuvo la implementación del sistema, dando solución a los requisitos funcionales identificados para resolver el problema de la investigación.
- ❖ Para evaluar la calidad del sistema se validó el cumplimiento de los requisitos definidos con las pruebas de caja negra utilizando la técnica de particiones de equivalencia, pruebas de seguridad y las pruebas de rendimiento. Finalmente se mostraron los resultados obtenidos durante la realización de las pruebas, para determinar que el software posee una calidad aceptable.

CONCLUSIONES GENERALES

Luego de culminado el desarrollo de la presente solución y dándole cumplimiento al objetivo general planteado, se obtienen las siguientes conclusiones:

- ❖ El estudio del marco conceptual permitió caracterizar las distintas aplicaciones que generan libros electrónicos en la actualidad, las cuales posibilitaron identificar la herramienta más idónea a utilizar en la Aplicación.
- ❖ Se realizó el análisis y diseño de la aplicación obtenido como resultado los artefactos y diagramas generados en todas las fases para el correcto desarrollo del mismo.
- ❖ La Aplicación para la Generación de Libros Web 2.0 se implementó de acuerdo a los 16 requisitos funcionales identificados en la fase de análisis y diseño.
- ❖ El diseño y ejecución de pruebas del sistema permitieron comprobar el correcto funcionamiento de la Aplicación para la Generación de Libros Web 2.0.

RECOMENDACIONES

Luego de analizado los resultados del presente trabajo de diploma, se puede llegar a las siguientes recomendaciones:

- ❖ Agregar funcionalidades que permitan modificar las plantillas predeterminadas, así como un motor generador que las valide de manera automática.

REFERENCIAS BIBLIOGRÁFICAS

1. **Ponce, Johnn Calvopiña.** Blog:Contenidos de Cultura informática. [En línea] 26 de Abril de 2012. [Citado el: 18 de Febrero de 2013.] <http://johnnjc.blogspot.com/2012/04/que-es-una-aplicacion-informatica.html>.
2. **Mora, Sergio Lujan.** *Programación de aplicaciones web* . España : s.n., 2010. ISBN-9788484542063 .
3. **Menéndez, Asensio Rafael.** www.um.es. [En línea] Commons 3.5, 10 de Agosto de 2012. [Citado el: 18 de febrero de 2013.] <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>.
4. **Suárez, Santiago Blanco.** roble.pntic.mec.es. [En línea] roble.pntic, Enero de 1999. [Citado el: 18 de Febrero de 2013.] <http://roble.pntic.mec.es/~sblanco1/libros1.htm>.
5. **Martínez, Yoandy Doble-Marcos.** *Trabajo de Diploma: Aplicación para la Generación de Libros Web.* Habana : UCI, 2011.
6. **McLure, Scott.** jutoh. [En línea] 27 de Junio de 2012. [Citado el: 18 de Febrero de 2013.] <http://jutoh.malavida.com/>.
7. **Varonas, Nico.** www.descargas.net. [En línea] 17 de Enero de 2011. [Citado el: 18 de Febrero de 2013.] <http://www.descargas.net/sigil-editor-ebook-libre-gratuito-multiplataforma/>.
8. **www.sourcefabric.org.** sourcefabric. [En línea] 2012. [Citado el: 18 de Febrero de 2013.] <https://www.sourcefabric.org/en/booktype/>.
9. **Hughes, Paul.** es.download.cnet.com. [En línea] 1 de Diciembre de 2009. [Citado el: 18 de Febrero de 2013.] http://es.download.cnet.com/Scrivener/3000-2079_4-139287.html.
10. **Epingsoft** . www.softpedia.es. [En línea] 16 de Enero de 2013. [Citado el: 18 de Febrero de 2013.] <http://www.softpedia.es/programa-ePub-Maker-157673.html>.
11. **Adobe.com.** www.nousmedis.com. [En línea] 10 de Enero de 2010. [Citado el: 18 de Febrero de 2013.] <http://www.nousmedis.com/2010/01/creacion-de-libros-electronicos-en-formato-epub-con-indesign-cs4/>.
12. **softonic.** reader-ebook-wizard.m.softonic.com. [En línea] 11 de Octubre de 2003. [Citado el: 18 de Febrero de 2013.] <http://reader-ebook-wizard.m.softonic.com/>.
13. **Sphinx.com.** sphinx-doc.org. *sphinx-doc.org.* [En línea] 10 de Marzo de 2012. [Citado el: 18 de Febrero de 2013.] <http://sphinx-doc.org/changes.html>.

14. **Romero, Hermenegildo.** [www.slideshare.net](http://www.slideshare.net/MeneRomero/metodologias-de-desarrollo). [En línea] 7 de Febrero de 2012. [Citado el: 18 de Febrero de 2013.] <http://www.slideshare.net/MeneRomero/metodologias-de-desarrollo>.
15. **Barradas, Yuliet.** [blogspot.com](http://yuliethbarradas.blogspot.com/2010/04/metodologia-agil-openup.html). [En línea] 21 de Abril de 2010. [Citado el: 18 de Febrero de 2013.] <http://yuliethbarradas.blogspot.com/2010/04/metodologia-agil-openup.html>.
16. **PRAT TERRENCE, MARVIN TZELKOWITZ.** *LENGUAJES DE PROGRAMACION: DISEÑO E IMPLEMENTACION*. MEXICO : PRENTICE HALL, 1999-02-10. 0-13-678012-1.
17. **ZAKAS, NICHOLAS C.** *JAVA SCRIPT PARA DESARROLLADORES WEB*. Madrd : WROX, 2006-07-10. 84-415-1953-6.
18. **Angel Cobo, Patricia Gomez, Daniel Perez.** *Tecnologías para el desarrollo de aplicaciones web*. España : Diaz de Santos(ww.diazdesantos.es/ediciones), 2005. 84-7978-706-6.
19. **Eguiluz, Javier.** [www.maestrosdelweb.com](http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/). [En línea] 5 de Septiembre de 2007. [Citado el: 20 de Febrero de 2013.] <http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>.
20. **librosweb.es.** [librosweb.](http://www.librosweb.es/symfony/index.html) [En línea] 2013. [Citado el: 19 de Febrero de 2013.] <http://www.librosweb.es/symfony/index.html>.
21. **Rosas, Juan Eladio Sánchez.** [blogs.antartec.com](http://blogs.antartec.com/desarrolloweb/tag/extjs/). [En línea] 28 de Octubre de 2008. [Citado el: 20 de Febrero de 2013.] <http://blogs.antartec.com/desarrolloweb/tag/extjs/>.
22. **Maribel.** [estudioteca.net](http://www.estudioteca.net/universidad/telecomunicaciones/gestor-base-datos/). [En línea] 21 de Marzo de 2012. [Citado el: 19 de Febrero de 2013.] <http://www.estudioteca.net/universidad/telecomunicaciones/gestor-base-datos/>.
23. **postgresql.org.** [postgresql.org](http://www.postgresql.org/docs/9.1/static/intro-what-is.html). [En línea] 2013. [Citado el: 19 de Febrero de 2013.] <http://www.postgresql.org/docs/9.1/static/intro-what-is.html>.
24. **Blanco, Carlos.** [carlosblanco.pro](http://carlosblanco.pro/2012/04/entornos-desarrollo-integrado-introduccion/). [En línea] 8 de Abril de 2012. [Citado el: 19 de Febrero de 2013.] <http://carlosblanco.pro/2012/04/entornos-desarrollo-integrado-introduccion/>.
25. **Ramirez, Ivan.** netbeans-ide.softonic.com. [En línea] 22 de Octubre de 2012. [Citado el: 20 de Febrero de 2013.] <http://netbeans-ide.softonic.com/>.
26. **Guillén, Paola Romero.** [www.itlalaguna.edu.mx](http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%C3%B1o%20orientado%20a%20objetos/Ch4intro.pdf). [En línea] [Citado el: 19 de Febrero de 2013.] <http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%C3%B1o%20orientado%20a%20objetos/Ch4intro.pdf>.
27. **Jimenez, Ing. Eliseo Castro.** [www.slideshare.net](http://www.slideshare.net/ecastrojimenez/uml-lenguaje-de-modelamiento-unificado-presentation). [En línea] 28 de Noviembre de 2008. [Citado el: 19 de Febrero de 2013.] <http://www.slideshare.net/ecastrojimenez/uml-lenguaje-de-modelamiento-unificado-presentation>.

28. **Meza, Mirna.** herramientascas. [En línea] 2 de Abril de 2011. [Citado el: 19 de Febrero de 2013.] <http://fds-herramientascas.blogspot.com/>.
29. **www.visual-paradigm.com.** www.visual-paradigm.com. [En línea] 1 de Mayo de 2012. [Citado el: 18 de Febrero de 2013.] <http://www.visual-paradigm.com/product/vpuml/editions/enterprise.jsp>.
30. **Larman, Craig.** *UML y Patrones.* Mexico : Cbara Nacional de la Industria Editorial Mexicana, 1999. ISBN 0-13-748880-7.
31. **Somerville, IAN.** Ingeniería del Software. [En línea] [Citado el: 21 de Febrero de 2013.] <http://www.filecrop.com/ian-sommerville-ingenieria-de-software-septima-edicion.pdf.html>.
32. **Rumbaugh, J, Jacobson, I y Booch, G.** *El Lenguaje Unificado de Modelado.* 2000.
33. **Pressman, Roger.** Ingeniería del Software 6ta edición. [En línea] [Citado el: 20 de Febrero de 2013.] <http://ebookbrowse.com/roger-s-pressman-ingenieria-del-software-v-ed-pdf-d47183684>.
34. **Ullman, Jeffrey y Widom, Jennifer.** *Introducción a los Sistemas de Bases de Datos.* . México 1999 : Editorial Prentice Hall, 1999. ISBN: 970-17-0256-5. .
35. **ROSA, DANIELA ENRIQUEZ.** mood.itdurango.edu.mx. [En línea] 14 de Marzo de 2009. [Citado el: 28 de Febrero de 2013.] <http://mood.itdurango.edu.mx/mod/forum/discuss.php?d=50>.
36. **Pressman, Roger.** *Ingeniería de Software: Un enfoque Práctico.* Madrid : España, 2001. ISBN.
37. **Roque, Diana Monne.** *Trabajo de Diploma: Aplicación de las pruebas de liberación al Sistema Informático de Genética Médica.* Cuba : s.n., 2008.

BIBLIOGRAFÍA

Adobe.com. www.nousmedis.com. [En línea] 10 de Enero de 2010. [Citado el: 18 de Febrero de 2013.] <http://www.nousmedis.com/2010/01/creacion-de-libros-electronicos-en-formato-epub-con-indesign-cs4/>.

Angel Cobo, Patricia Gomez, Daniel Perez. *Tecnologías para el desarrollo de aplicaciones web.* España : Diaz de Santos(ww.diazdesantos.es/ediciones), 2005. 84-7978-706-6.

Bakken, Stig Sæther. Manual_de_PHP. [En línea] <http://php.uci.cu/search.php>.

Barradas, Yuliet. blogspot.com. [En línea] 21 de Abril de 2010. [Citado el: 18 de Febrero de 2013.] <http://yuliethbarradas.blogspot.com/2010/04/metodologia-agil-openup.html>.

Blanco, Carlos. carlosblanco.pro. [En línea] 8 de Abril de 2012. [Citado el: 19 de Febrero de 2013.] <http://carlosblanco.pro/2012/04/entornos-desarrollo-integrado-introduccion/>.

Booch, G., Jacobson, I. y Rumbaugh, J. *The UML specification documents.* s.l. : Rational Software Corp., 1997.

Camacho, E., Fabio, C. y Gabriel, N. Arquitecturas de Software. Guía de estudio. [En línea] 2004. [Citado el: 27 de 02 de 2013.] <http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf>.

Catalani, Exequiel. “Arquitectura Modelo/Vista/Controlador”. “*Arquitectura Modelo/Vista/Controlador*”. [En línea] [Citado el: 27 de 02 de 2013.] <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>.

Craig Larman. UML y Patrones. [En línea] www.publidisa.com/preview-libro-9788483229279.pdf desventajas.html

Eguiluz, Javier. www.maestrosdelweb.com. [En línea] 5 de Septiembre de 2007. [Citado el: 20 de Febrero de 2013.] <http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>.

Epingsoft . www.softpedia.es. [En línea] 16 de Enero de 2013. [Citado el: 18 de Febrero de 2013.] <http://www.softpedia.es/programa-ePub-Maker-157673.html>.

ExtJS. [En línea] <http://www.sencha.com/products/extjs3/>.

Fabien Potencier. Symfony la Guía Definitiva. [En línea]

García Mato, Rosa María. *Diseño de bases de datos.* 1999. pág. 51.

Guillén, Paola Romero. www.itlalaguna.edu.mx. [En línea] [Citado el: 19 de Febrero de 2013.] <http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%C3%B1o%20orientado%20a%20objetos/Ch4intro.pdf>.

Guillermo Lemus. Tipos de pruebas del software <http://www.slideshare.net/GuillermoLemus/tipos-de-pruebas-de-software>

Hughes, Paul. es.download.cnet.com. [En línea] 1 de Diciembre de 2009. [Citado el: 18 de Febrero de 2013.] http://es.download.cnet.com/Scrivener/3000-2079_4-139287.html.

IAN Somerville. Ingeniería del Software. [En línea] <http://www.filecrop.com/ian-sommerville-ingenieria-de-software-septima-edicion.pdf.html>.

Ingeniería de Software: Un enfoque Práctico. Madrid : s.n., 2001.

International, Visual Paradigm. Visual Paradigm. *Visual Paradigm.* [En línea] [Citado el: 25 de 02 de 2013.]

Jimenez, Ing. Eliseo Castro. www.slideshare.net. [En línea] 28 de Noviembre de 2008. [Citado el: 19 de Febrero de 2013.] <http://www.slideshare.net/ecastrojimenez/uml-lenguaje-de-modelamiento-unificado-presentation>.

Larman, Craig. *UML y Patrones.* Mexico : Cbara Nacional de la Industria Editorial Mexicana, 1999. ISBN 0-13-748880-7.

librosweb.es. librosweb. [En línea] 2013. [Citado el: 19 de Febrero de 2013.] <http://www.librosweb.es/symfony/index.html>.

Maribel. estudianteca.net. [En línea] 21 de Marzo de 2012. [Citado el: 19 de Febrero de 2013.] <http://www.estudianteca.net/universidad/telecomunicaciones/gestor-base-datos/>.

Martinez, Rafael. Sobre Postgresql. [En línea] http://www.postgresql.org.es/sobre_postgresql.

McLure, Scott. jutoh. [En línea] 27 de Junio de 2012. [Citado el: 18 de Febrero de 2013.] <http://jutoh.malavida.com/>.

Meza, Mirna. herramientascas. [En línea] 2 de Abril de 2011. [Citado el: 19 de Febrero de 2013.] <http://fds-herramientascas.blogspot.com/>.

MONTALVO, C. Ventajas de las aplicaciones web, 26 de Octubre de 2008.

Pérez, Javier Eguíluz. Manual de java script. [En línea] www.librosweb.es.

postgresql.org. postgresql.org. [En línea] 2013. [Citado el: 19 de Febrero de 2013.] <http://www.postgresql.org/docs/9.1/static/intro-what-is.html>.

PRAT TERRENCE, MARVIN TZELKOWITZ. *LENGUAJES DE PROGRAMACION: DISEÑO E IMPLEMENTACION.* MEXICO : PRENTICE HALL, 1999-02-10. 0-13-678012-1.

Pressman, Roger. Ingeniería del Software 6ta edición. [En línea] [Citado el: 20 de Febrero de 2013.] <http://ebookbrowse.com/roger-s-pressman-ingenieria-del-software-v-ed-pdf-d47183684>.

Ramirez, Ivan. netbeans-ide.softonic.com. [En línea] 22 de Octubre de 2012. [Citado el: 20 de Febrero de 2013.] <http://netbeans-ide.softonic.com/>.

Repositorio de Documentación <http://docs.prod.uci.cu/ext-3.3.1/>

Riggs, S. y Krosing, H. *PostgreSQL 9. Administration Cookbook.* Birmingham, E.E. U.U. : Packt Publishing, 2010.

Romero, Hermenegildo. www.slideshare.net. [En línea] 7 de Febrero de 2012. [Citado el: 18 de Febrero de 2013.] <http://www.slideshare.net/MeneRomero/metodologias-de-desarrollo>.

Roque, Diana Monne. *Trabajo de Diploma: Aplicación de las pruebas de liberación al Sistema Informático de Genética Médica.* Cuba : s.n., 2008.

Rosas, Juan Eladio Sánchez. blogs.antartec.com. [En línea] 28 de Octubre de 2008. [Citado el: 20 de Febrero de 2013.] <http://blogs.antartec.com/desarrolloweb/tag/extjs/>.

SCHNEIER, B. Vulnerabilidades, 15 de Septiembre de 2000. [Disponible en: <http://www.schneier.com/crypto-gram-0009.htm>]

softonic. reader-ebook-wizard.m.softonic.com. [En línea] 11 de Octubre de 2003. [Citado el: 18 de Febrero de 2013.] <http://reader-ebook-wizard.m.softonic.com/>.

Somerville, IAN. Ingeniería del Software. [En línea] [Citado el: 21 de Febrero de 2013.] <http://www.filecrop.com/ian-sommerville-ingenieria-de-software-septima-edicion.pdf.html>.

Sphinx.com. sphinx-doc.org. *sphinx-doc.org.* [En línea] 10 de Marzo de 2012. [Citado el: 18 de Febrero de 2013.] <http://sphinx-doc.org/changes.html>.

Suárez, Santiago Blanco. roble.pntic.mec.es. [En línea] roble.pntic, Enero de 1999. [Citado el: 18 de Febrero de 2013.] <http://roble.pntic.mec.es/~sblanco1/libros1.htm>.

Varonas, Nico. www.descargas.net. [En línea] 17 de Enero de 2011. [Citado el: 18 de Febrero de 2013.] <http://www.descargas.net/sigil-editor-ebook-libre-gratuito-multiplataforma/>.

www.buenastareas.com. [En línea] buenastareas, Mayo de 2010. [Citado el: 18 de Febrero de 2013.] <http://www.buenastareas.com/ensayos/Definici%C3%B3n-y-Tipos-De-Aplicaciones-Web/317130.html>.

www.e-lemental.com. [En línea] e-lemental, Septiembre de 2010. [Citado el: 18 de febrero de 2013.] <http://www.e-lemental.com.pe/que-es-una-aplicacion-web/>.

www.mastermagazine.inf. [En línea] mastermagazine. [Citado el: 18 de 2 de 2013.] <http://www.mastermagazine.info/termino/3874.php#ixzz2HmdEY7be>.

www.sourcefabric.org. sourcefabric. [En línea] 2012. [Citado el: 18 de Febrero de 2013.] <https://www.sourcefabric.org/en/booktype/>.

www.visual-paradigm.com. www.visual-paradigm.com. [En línea] 1 de Mayo de 2012. [Citado el: 18 de Febrero de 2013.] <http://www.visual-paradigm.com/product/vpuml/editions/enterprise.jsp>.

ZAKAS, NICHOLAS C. *JAVA SCRIPT PARA DESARROLLADORES WEB*. Madrd : WROX, 2006-07-10. 84-415-1953-6.

ANEXOS

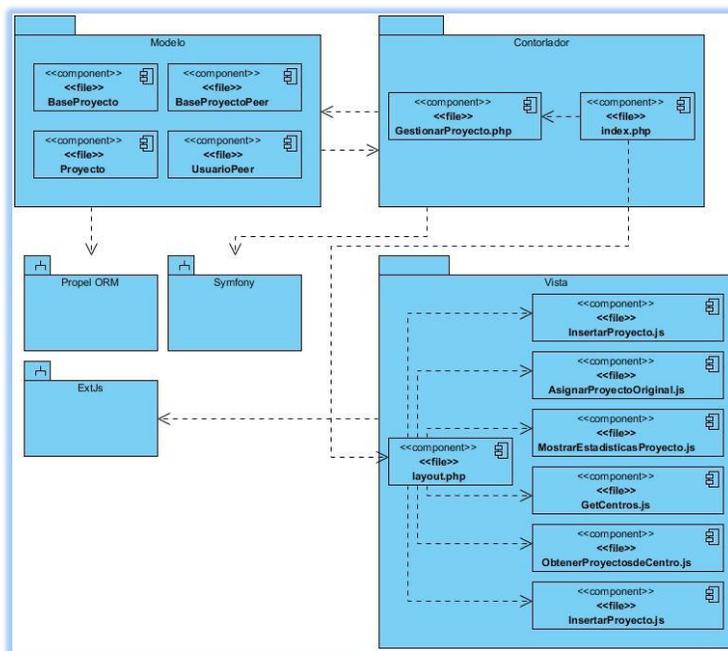


Figura 19: Anexo 1: Diagrama de componente del CU: Gestionar Proyecto

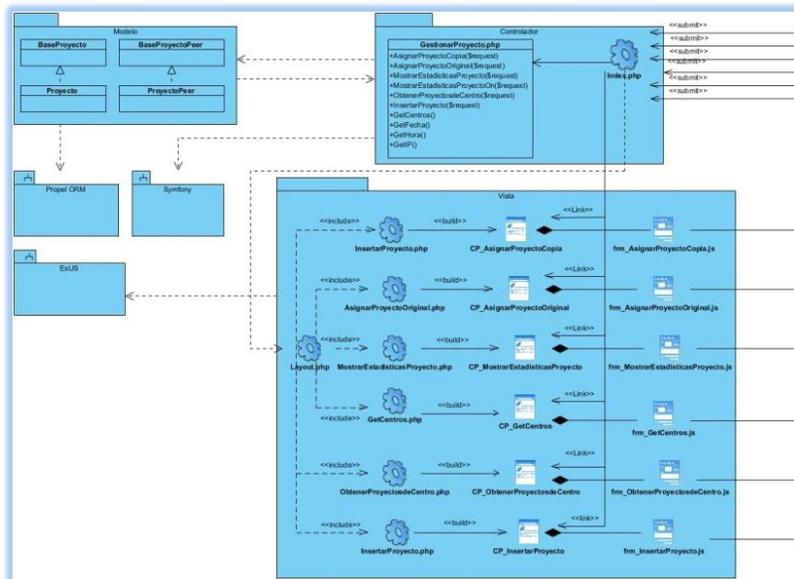


Figura 200: Anexo 2: Diagrama de clases del diseño del CU Gestionar Proyecto



Figura 211: Anexo 3: Interfaz principal de la Aplicación.

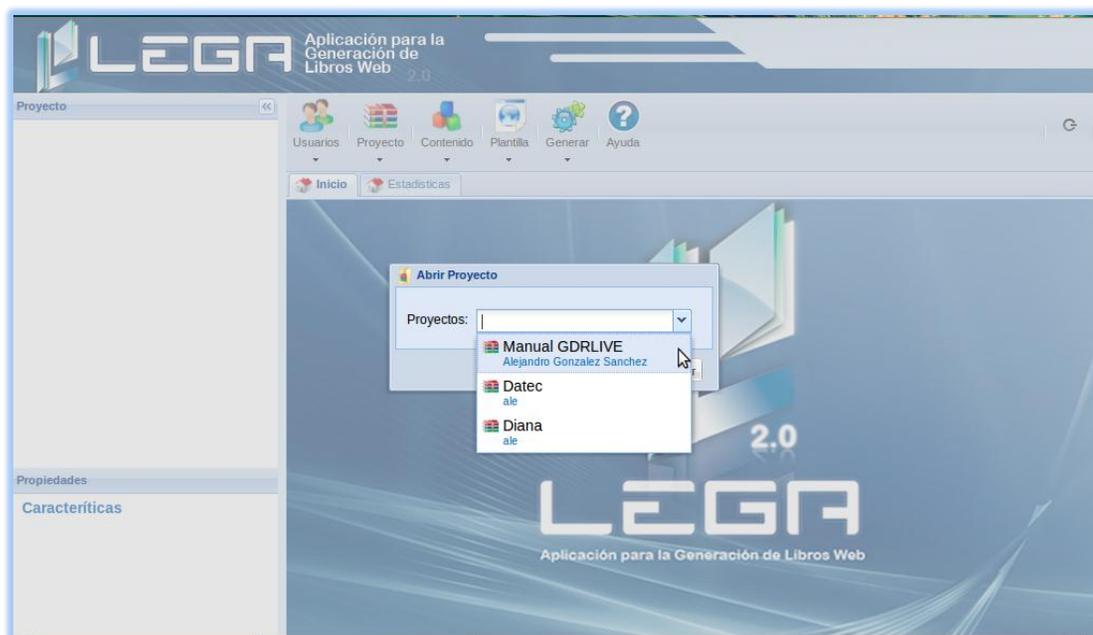


Figura 222: Anexo 4: Interfaz principal, funcionalidad abrir proyecto

GLOSARIO

PLW: Proyecto Libro Web.

HTML: Lenguaje de Marcado de Hipertexto (*en inglés Hypertext Markup Language*)

CSS: Hojas de Estilo en Cascada (*en inglés Cascading Style Sheets*)

PHP: Acrónimo recursivo que significa *PHP Hypertext Pre-processo*.

API: Una interfaz de programación de aplicaciones (*en inglés application programming interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usados generalmente en las bibliotecas.

HTTP: Protocolo de transferencia de hipertexto (*en inglés Hypertext Transfer Protocol*)

GNU: Licencia Pública General (*en inglés General Public License*)

BD: Base de datos (*en inglés Data Base*)

IDE: Entorno de desarrollo integrado (*en inglés Integrated Developed Environment*)

CASE: .Conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de Software y desarrolladores (*en inglés Cumputer Aided Software Engineering*)

CRUD: Su nombre es un acrónimo de las palabras en inglés Create, Read, Update, Delete.

MVC: Modelo Vista Controlador.(*en inglés Model, View and Controler*)

TIC: Tecnologías de la Información y la Comunicación

EPUB: Publicación electrónica (*en inglés Electronic publication*)

MOBIPOCKET: Formato para archivos de libro electrónico (*e-book*)

WYSIWYG: "Lo que ves es lo que obtienes" (*en inglés What You See Is What You Get*)... 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,

MAC OS X: La definición consiste en una serie de sistemas operativos basados en Unix