Universidad de las Ciencias Informáticas

Facultad 6

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Título: Sistema de Gestión de Contenidos para la Generación de Libros Web

Autor:

Adler Mármol Rodríguez

Tutores:

Ing. Lianet Cabrera González

Ing. Ernesto Dueñas Rodríguez

“I don't care that they stole my idea... I care that they don't have any of their own”.

Nikola Tesla

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis que tiene por título: Sistema de Gestión de Contenidos para la Generación de Libros Web 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 ____.

Adler Mármol Rodríguez

_____

Firma del Autor

Ing. Ernesto Dueñas Rodríguez

_____

Firma del Tutor

Ing. Lianet Cabrera González

_____

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Lianet Cabrera González

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Años de experiencia en el tema: 3

Años de graduado: 3

e-mail: **lcabrerag@uci.cu**

Tutor: Ing. Ernesto Dueñas Rodríguez

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Años de experiencia en el tema: 3

Años de graduado: 3

e-mail: **eduenas@uci.cu**

DEDICATORIA

A mis padres Grover y María.

A mis amigos y a mis tutores (amigos también).

A Rebeca.

RESUMEN

El Centro de Tecnologías de Gestión de Datos (DATEC) provee variadas soluciones integrales y servicios especializados en el análisis y el almacenamiento de datos. Para hacer efectivo el empleo de las aplicaciones informáticas que son desarrolladas en dicho centro, se precisa orientar a los usuarios en el trabajo con cada una de ellas a través de manuales de ayuda. Actualmente, la confección de estos manuales para las aplicaciones web desarrolladas en DATEC se lleva a cabo empleando herramientas de edición de textos como Open Office u otros similares. Como resultado, los manuales obtenidos no presentan una uniformidad de formato y de diseño que identifique a los productos desarrollados en DATEC. Es por esto y por la necesidad en algunos casos de integrarlos como parte de las soluciones para mayor portabilidad y accesibilidad, que los analistas del centro emplean su contenido para redactar libros web que pueden ser embebidos en las soluciones desarrolladas, para de esta manera facilitar la asistencia a los usuarios. La propuesta de solución consiste en un sistema de gestión de contenidos para la creación de libros web, que permita a los usuarios la generación de manuales de ayuda. La solución obtenida como resultado de la investigación cuenta con una interfaz intuitiva y un conjunto de funcionalidades que garantizan crear, editar y/o exportar los manuales creados en formato de libro web con formato y diseño uniformes. Además, se validó a través de varios tipos de pruebas para asegurar su correcto funcionamiento y la satisfacción de los usuarios.

Palabras Claves: gestión de contenidos, libro web, manual de ayuda.

ABSTRACT

The Technology Centre Data Management (DATEC) provides various solutions and specialized services in analysis and data storage. To make effective use of applications that are developed at the center, it is necessary to guide users in working with each of them through help manuals. Currently, the preparation of these manuals for Web applications developed in DATEC is done using text editing tools such as Open Office or similar. As a result, manual obtained do not have a uniform format and design that identifies the products developed in DATEC. It is because of this and the need in some cases to integrate them as part of the solutions for portability and accessibility, analysts from the center use their content to write web books that can be embedded solutions developed, to thereby facilitate assistance to users. The proposed solution is a Content Management System for building web books, allowing users to generate help manuals. The solution obtained as a result of the research has an intuitive interface and a set of features that guarantee create, edit and export manuals created in web book format, with an uniform format and design. It was further validated through several types of tests to ensure proper performance and user satisfaction.

Key Words: *content management system, help manual, web book.*

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En la era de la información constituye una necesidad primaria la transmisión (cada vez más rápida y de mayor alcance) de todo el conocimiento que constantemente adquiere el hombre. Una de las alternativas para alcanzar este fin lo constituyen los llamados libros electrónicos.

Un libro electrónico no es más que la versión digital de un libro convencional, de manera que puede ser almacenado y leído en dispositivos electrónicos. Puede ser transmitido a través de las redes de manera sencilla y su distribución no requiere el empleo de materias primas. Estas características lo convierten en una alternativa más económica que las publicaciones de texto en revistas, periódicos y libros convencionales. Los libros electrónicos pueden ser publicados en diferentes formatos de acuerdo a la herramienta donde serán leídos, uno de estos formatos lo constituye el libro web. Un libro web consta de una o más páginas web, cada una de ellas relacionadas entre sí por medio de hipervínculos, de manera similar a un sitio web.

Por otro lado, el auge de la automatización de procesos y servicios, buscando satisfacer necesidades en la población, dio lugar a la creación de innumerables sistemas informáticos destinados a disímiles usuarios. Pero el desconocimiento y la inexperiencia en el empleo de dichos sistemas dificultaban su uso efectivo, dando lugar al surgimiento del manual de ayuda como elemento fundamental para comprender el funcionamiento de un software.

Un manual de ayuda de un software es un documento de comunicación técnica que muestra los procesos que el usuario debe realizar con el sistema informático. Permite conocer en detalle qué actividades deberá desarrollar para ejecutar las funcionalidades del software. Agrupa la información, las normas y la documentación necesaria para que el usuario conozca y utilice adecuadamente la aplicación desarrollada.

En Cuba, el uso de los manuales de ayuda constituye un paso de avance en cuanto al trabajo con aplicaciones informáticas. El Centro de Tecnologías de Gestión de Datos (DATEC), perteneciente a la Universidad de las Ciencias Informáticas (UCI), provee soluciones integrales y servicios informáticos especializados en el análisis y el almacenamiento de datos. Para la utilización de las aplicaciones que desarrolla dicho centro, se precisa orientar al usuario en el trabajo con cada una de ellas a través de un manual de ayuda.

Para la elaboración de dichos manuales se emplean herramientas de terceros tales como Microsoft Word o Libre Office. Como resultado los manuales existentes se encuentran en un formato compatible con dichas herramientas y no existe uniformidad en el diseño de los mismos.

Con el fin de publicar los manuales de ayuda como parte de las aplicaciones desarrolladas, los analistas del centro emplean el contenido de los manuales existentes para crear libros web. Para lograr este fin las herramientas empleadas actualmente presentan una serie de limitantes que a continuación se enumeran:

- En algunos casos, no permiten exportar el contenido en formato web.
- No permiten incorporar textos cuyo formato haya sido previamente definido a través de herramientas de edición de texto (Microsoft Word o Libre Office).
- Es limitado el trabajo con los contenidos de un manual (imágenes, tablas, textos e hipervínculos).

Estas limitantes traen consigo que la confección de un libro web se torne engorrosa. Además de que la selección de una herramienta para crear un libro web, y su posterior comprensión, propician la pérdida de tiempo valioso en el proceso de desarrollo de software.

Por lo anteriormente expuesto se plantea como **problema de la investigación** ¿Cómo garantizar la adecuada gestión de contenidos para la generación libros web?, definiéndose como **objeto de estudio** el proceso de gestión de contenidos, enmarcado en el **campo de acción** herramientas para la generación de libros web.

Para darle solución al problema se traza como **objetivo general** desarrollar un Sistema de Gestión de Contenidos para la Generación de Libros Web (SGCLW) en el centro DATEC y en correspondencia con él, los siguientes **objetivos específicos:**

- Analizar los conceptos, tecnologías y herramientas empleados actualmente para la generación de libros web.
- Realizar el análisis y el diseño del SGCLW.
- Implementar el SGCLW.
- Verificar el correcto funcionamiento del SGCLW.

Para guiar la lógica de la investigación se enumeran las siguientes **preguntas científicas**:

- ¿En qué se fundamenta el proceso de Gestión de Contenidos para la Generación de Libros Web?
- ¿Qué tecnologías, metodología y herramientas seleccionar para desarrollar el Sistema de Gestión de Contenidos para la Generación de Libros Web?
- ¿Qué características debe presentar el Sistema de Gestión de Contenidos para la Generación de Libros Web para satisfacer las necesidades del cliente?
- ¿Cómo estructurar el proceso de diseño e implementación del Sistema de Gestión de Contenidos para la Generación de Libros Web?

- ¿Cómo comprobar el correcto funcionamiento del Sistema de Gestión de Contenidos para la Generación de Libros Web?

Con el fin de resolver el problema de la investigación y dar respuesta a las preguntas antes planteadas se enumeran las siguientes **tareas de la investigación:**

- Análisis de los conceptos asociados al problema de la investigación.
- Selección de las tecnologías, metodología y herramientas a utilizar en el desarrollo del SGCLW.
- Identificación de los requisitos funcionales y no funcionales para el correcto funcionamiento del SGCLW.
- Confección del modelo de casos de uso SGCLW.
- Elaboración del modelo de diseño del SGCLW.
- Elaboración del modelo de implementación del SGCLW.
- Diseño de los casos de prueba para determinar el correcto funcionamiento del SGCLW.
- Identificación y resolución de las no conformidades identificadas en la ejecución de las pruebas.

Para orientar la investigación se aplicaron **métodos teóricos y empíricos**. Los métodos empíricos se utilizan para descubrir y acumular un conjunto de hechos y datos como base para verificar la hipótesis o dar respuesta a las preguntas científicas de la investigación, pero que no son suficientes para profundizar en las relaciones esenciales que se dan en los procesos. Los métodos teóricos posibilitan, a partir de los resultados obtenidos por medio de los métodos empíricos, sistematizarlos, analizarlos, explicarlos, descubrir qué tienen en común, para llegar a conclusiones confiables que permitan resolver el problema.

De los métodos empíricos se aplicó la observación para corroborar lo engorroso que resulta el empleo de las herramientas que emplean los analistas actualmente para generar libros web. Este método permitió descartar el empleo de otras herramientas que pudieran ser aplicables pero no se adecúan a las características del problema.

Se hizo empleo de los métodos teóricos que a continuación se enumeran:

- Se utilizó el análisis-síntesis para sistematizar toda la información relacionada con los libros electrónicos y de la utilización que se le dan a los manuales de ayuda. Se realizó un estudio de los lenguajes de programación requeridos para implementar la aplicación, del uso de las tecnologías para aplicarlas a la solución, así como la selección de la metodología a seguir durante el desarrollo del sistema.

- Se utilizó el método inductivo-deductivo para determinar un conocimiento sobre la realidad. Este método permitió conocer cómo se lleva a cabo la generación de manuales de ayuda en el departamento actualmente.

El presente trabajo de diploma está estructurado de la siguiente manera:

- **CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN**

  Se describen aspectos teóricos muy elementales referentes a la generación de libros web para argumentar la presente investigación. A partir del análisis de los aspectos teóricos descritos, se hace una propuesta de las herramientas y tecnologías a utilizar durante el desarrollo de la aplicación con el fin de dar respuesta a la situación problemática plateada.

- **CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN**

  Se basa principalmente en el diseño de las funcionalidades para la elaboración del sistema. A través del modelo de dominio se muestra el problema que existe en el centro DATEC. Para darle solución al mismo se definen requisitos del sistema. Los requisitos funcionales son agrupados en casos de usos y se presenta una descripción detallada del caso de uso arquitectónicamente significativo del sistema. Se identifican los patrones de diseño empleados durante la construcción del sistema. Finalmente se propone el modelo de despliegue a emplear para poner en funcionamiento la aplicación.

- **CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA**

  Se muestra el modelo de implementación como resultado del diseño llevado a cabo anteriormente. Su propósito es completar el desarrollo del sistema basado en la arquitectura definida, haciendo uso del diseño del diagrama de componentes de la aplicación. Al mismo tiempo se comprueba que el producto final genere correctamente manuales de ayuda con la calidad requerida. El funcionamiento del software es evaluado a través de las pruebas establecidas durante este capítulo.

Posteriormente se presentan conclusiones generales, recomendaciones y referencias bibliográficas para un mejor entendimiento de los resultados de la investigación.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

Introducción

Con el objetivo de comprender los conceptos fundamentales asociados actualmente a la generación de manuales de ayuda, en el presente capítulo se describen los aspectos teóricos fundamentales asociados al dominio del problema planteado. Se lleva a cabo un estudio de las herramientas empleadas actualmente para la generación de manuales de ayuda y libros electrónicos en general. A partir del estudio de dichos aspectos, se selecciona la metodología de desarrollo de software a seguir durante el proceso de desarrollo de la solución. Finalmente se hace una propuesta de las herramientas y tecnologías a utilizar durante el desarrollo de la aplicación, con el fin de dar respuesta a la situación problemática planteada.

1.1 Libro electrónico

"Un libro electrónico no es otra cosa que la versión digital de un libro de papel, de tal forma que puede visualizarse en cualquier dispositivo digital ya sean ordenadores, teléfonos móviles, lectores de libros electrónicos o Ipads[1]" (1). Por lo general se generan en formato de publicación electrónica (*epub*[2]). Se utiliza como medio digital para presentar información utilizando diferentes recursos tales como textos, gráficos e imágenes.

El libro electrónico provee grandes ventajas con respecto a los libros convencionales para el mundo tecnológico actual, algunas de ellas son:

- Facilidad de publicación y lectura. Cada usuario puede leerlos desde sus ordenadores u otro dispositivo digital que esté al alcance.
- Portabilidad y facilidad de consulta.
- Son mucho más económicos que los libros impresos, ya que no es necesario el empleo de materias primas para su confección y distribución.

El contenido de un libro electrónico puede distribuirse de igual manera que el contenido asociado a un libro común, por medio de secciones que constituyen capítulos compuestos de epígrafes. La variedad de la exposición en la información incrementa la motivación del lector. Por ello, se suele combinar en cada sección recursos como texto e imágenes para estimular la comprensión del lector sobre el contenido.

[1] El *iPad* es una línea de tabletas electrónicas diseñadas y comercializadas por Apple Inc.
[2] *Epub* es una extensión de fichero en la informática, sus siglas provienen de "*electronic publication*", en español publicación electrónica.

1.1.1 Libro web

Existe una variante de libro electrónico que beneficia el trabajo del usuario con el mismo a través de la navegación. El usuario puede realizar una navegación más cómoda a través de un menú que representa el árbol de contenidos y que contiene vínculos para acceder a las diferentes secciones. Cada sección del libro está contenida en una página en formato HTML[3]. Esta variante de libro electrónico se conoce como libro web y aporta las características que a continuación se enumeran:

- Permite la navegación a través de las diferentes secciones del libro haciendo empleo de hipervínculos.
- Se puede enriquecer el contenido de las secciones del libro con contenidos como videos y sonidos.
- Puede ser integrado como parte de un sitio web.
- Un libro web publicado en la red es accesible en tiempo real.

1.2 Manual de ayuda

"Un manual de ayuda (o manual de usuario) es un conjunto de técnicas de uso y recomendaciones que acompañan a un producto en el momento de la adquisición para que el usuario pueda hacer un uso adecuado y eficiente del mismo (…) Puede contener varios idiomas si es un producto de venta internacional, y su redacción debe ser sencilla y didáctica" (2). El manual de ayuda de un software incluye capturas de pantalla de diferentes escenarios de la aplicación, así como diagramas claramente detallados y sencillos, que enumeran los pasos a realizar por el usuario para llevar a cabo las funcionalidades disponibles.

El objetivo del manual de ayuda es asesorar al usuario para que realice las siguientes funciones:

- Conocer cómo preparar los datos de entrada, cómo obtener resultados y los datos de salida.
- Servir como manual de aprendizaje y de referencia.
- Definir las funciones que debe realizar el usuario e informarle qué respuesta debe dar ante cada posible mensaje de error.
- Definir el orden de tratamiento del programa.
- Conocer las operaciones de entrada y salida de los datos, así como los resultados de las operaciones realizadas a partir de los datos introducidos (2).

[3]HTML, siglas de *HyperText Markup Language*, hace referencia al lenguaje de marcado para la elaboración de páginas web.

1.3 Herramientas empleadas para la construcción de manuales de ayuda

Generalmente un manual de ayuda perteneciente a un software contiene grandes cantidades de información estructuradas en secciones. De esta manera el usuario puede consultar la sección asociada a la funcionalidad del software que desea explotar, sin necesidad de leer el manual en su totalidad. Es posible representar un manual de ayuda en un libro electrónico común. Pero tanto la creación de la estructura del mismo, como la búsqueda de una de las secciones por parte de un usuario que lo consulta pueden tornarse engorrosas a medida que crece la cantidad de información contenida en el documento. Es por esto que se emplean herramientas que facilitan la confección de manuales de ayuda intuitivos y de fácil comprensión para los usuarios.

1.3.1 Robohelp

Robohelp es una herramienta informática destinada a la creación de manuales de ayuda. Esta herramienta es propiedad de *Adobe Systems*[4] *y* se distribuye bajo licencia privativa. Robohelp presenta las características siguientes:

- La dinámica está centrada en el usuario y el contenido del manual, lo que permite mejorar la experiencia del usuario final.
- Permite importar fácilmente el contenido de las aplicaciones de *Microsoft Word*, *Adobe* PDF y XML.
- Puede crear un sistema de información básico, que incluye secciones, tablas de contenido, glosarios, ayuda sensible al contexto y muchas otras opciones.
- Permite generar un sistema de referencia en todos los formatos populares (.HTML, .XML, .DOC, .PDF)
- Posibilita la preparación de la versión de impresión (3).

1.3.2 Dr. Explain

Dr. Explain es un software privativo que facilita la creación de archivos de ayuda, guías de usuario, manuales en línea y documentación de aplicaciones (4). Este programa analiza la aplicación y produce automáticamente capturas de pantalla de las ventanas junto con la secuencia de llamadas explicativas para cada control en la misma, basado en anotaciones en el código fuente de la aplicación a la que se le genera el manual de usuario. Esta herramienta cuenta además con las siguientes características:

[4] Empresa de software estadounidense que destaca en el mundo del software por sus programas de edición de páginas web, vídeos e imágenes.

- Posee una herramienta de captura de pantallas integrada, que analiza las ventanas o las páginas web de la aplicación y que automáticamente crea gráficos de las capturas de pantalla con anotaciones.
- Cuenta con un editor de contenidos con muchas funcionalidades, optimizado para crear documentación de software.
- Permite agregar funciones de búsqueda e índices de palabras clave a los manuales en línea sin necesidad de instalar programas, scripts ni bases de datos en el servidor.
- La creación de archivos de ayuda, de manuales en línea en formato HTML, de documentos RTF[5] y de documentación en formato PDF desde una única fuente.

A pesar de ser herramientas muy potentes y contar con funcionalidades útiles para la creación de manuales de ayuda; Robohelp y *Dr. Explain* cuentan con la limitante de ser software de tipo privativo. Las políticas de la universidad fomentan el empleo de software libre, descartando de esta manera el empleo de dichas herramientas.

1.4 Herramientas alternativas para la creación de manuales de ayuda

La creación de un manual de ayuda no se limita al empleo de un software destinado a este fin. Puede emplearse de igual manera una herramienta que permita la generación de libros electrónicos, para representar el contenido de un manual en las diferentes secciones de una publicación electrónica.

1.4.1 Sigil

Sigil es una herramienta diseñada para la edición de libros electrónicos con extensión *epub*. Es un software gratuito y de código abierto. Posee dos modos de uso: uno sencillo, básicamente para formatear el texto y otro avanzado para modificar el código del fichero donde se almacena el libro electrónico. Sigil posee además las características que a continuación se mencionan:

- Posee varios modos de visualización del contenido en edición: vista de libro, vista de código y presentación previa.
- Permite la revisión ortográfica de los textos con varios diccionarios.
- Es compatible con los sistemas operativos: Windows, MacOS y Linux (5).

[5] RTF, siglas de *Rich Text Format* es un formato de archivo informático desarrollado por Microsoft en 1987 para el intercambio de documentos multiplataforma. La mayoría de procesadores de texto son capaces de leer y escribir documentos de este tipo.

A pesar de ser una herramienta de código abierto y multiplataforma, Sigil es una herramienta de escritorio que debe ser instalada en cada ordenador donde se empleará. No permite la importación de contenido con formato previamente definido en los editores de texto más empleados. Estas últimas características descartan su empleo para la generación de manuales de ayuda en el departamento.

1.4.2 Datbook

Datbook es una herramienta para la generación de libros web desarrollada por uno de los departamentos del centro DATEC de la Universidad de las Ciencias Informáticas en Cuba. La aplicación define el uso de un motor de generación de documentación para obtener los libros web y puede ejecutarse en cualquier sistema operativo debido a que es una aplicación web. Esta herramienta no está en uso actualmente por un conjunto de deficiencias que a continuación se mencionan:

- La interfaz visual de la aplicación no es intuitiva.
- El trabajo con los recursos (tablas, imágenes, texto, hipervínculos) de un libro web es limitado.
- No permite incorporar texto con formato previamente definido en herramientas de edición de texto (Microsoft Word y Libre Office).
- No define un formato visual estándar para los manuales de ayuda.

Finalmente se optó por la construcción de una nueva herramienta que se adapte a las necesidades particulares del departamento Desarrollo de Aplicaciones. No se tomará en cuenta el uso de un motor generador de documentación debido a que estos realmente están destinados a la generación de documentación del código fuente de aplicaciones, y no de manuales de ayuda. Además, se propone hacer uso de un editor de texto HTML potente para gestionar el contenido de los libros web.

Las herramientas estudiadas poseen características que pueden ser aprovechadas como: la importación de contenido pre formateado desde Libre Office, la posibilidad de revisión ortográfica y la facilidad de búsqueda en el contenido del manual de ayuda.

1.5 Metodología para el desarrollo del software

Según Roger Pressman: "*una metodología es el conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevos productos de software (...)"* (6). Desarrollar un sistema no es tarea fácil y el uso de una metodología de desarrollo disminuye en gran medida la dificultad del programa. Las metodologías orientan y definen el ciclo de vida del desarrollo de un proyecto. Describen los pasos a seguir en cada etapa para obtener un producto eficiente y con calidad.

Debido a que el equipo de desarrollo destinado a la construcción del SGCLW cuenta con una sola persona: para dirigir el desarrollo del sistema se precisa del uso de metodologías ágiles. Las metodologías ágiles están especialmente preparadas para sufrir cambios durante el transcurso del proyecto. Permiten el desarrollo de un software en un corto período de tiempo y exigen poca documentación. Posibilitan un estrecho vínculo con el cliente y están diseñadas para equipos pequeños. Este tipo de metodología guiará el proceso de desarrollo para la solución del problema planteado.

Para el desarrollo de la aplicación se selecciona OpenUP como metodología a seguir debido a las características que presenta y a que es la metodología de desarrollo de software propuesta en la línea base de la arquitectura del departamento Desarrollo de Aplicaciones. Haciendo uso de esta metodología se disminuyen las probabilidades de fracaso e incrementa las probabilidades de éxito. OpenUP detecta errores en fechas tempranas del desarrollo a través de su ciclo iterativo, evitando la elaboración de documentos, diagramas e iteraciones innecesarias.

OpenUP

Es una metodología de desarrollo unificado basada en *Rational Unified Process* (RUP), dirigida por casos de uso y centrado en la arquitectura, basada en un progreso iterativo e incremental; apropiada para proyectos pequeños y de bajos recursos. Desarrolla un ciclo de vida interactivo que mitiga los riesgos a tiempo y ofrece demostrar resultados en curso al cliente del proyecto (7).

En la Figura 1 **¡Error! No se encuentra el origen de la referencia.**se muestra el ciclo de vida de la metodología OpenUP, que consta de 4 fases. La fase de **Inicio** se encarga del entendimiento del propósito y objetivos, adquiriendo suficiente información para confirmar qué debe hacer el proyecto. El objetivo es capturar las necesidades de los *stakeholders* en los objetivos del ciclo de vida para el desarrollo del *software*. La fase de **Elaboración** trata los riesgos más significativos para la arquitectura. Su propósito es establecer la base para la elaboración de la arquitectura del sistema. La fase de **Construcción** se enfoca al diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. Su propósito es completar el desarrollo del sistema basado en la arquitectura definida. Por último, está la fase de **Transición,** cuyo propósito es asegurar que el sistema se entrega a los usuarios, evaluando la funcionalidad del último entregable de la fase de construcción.

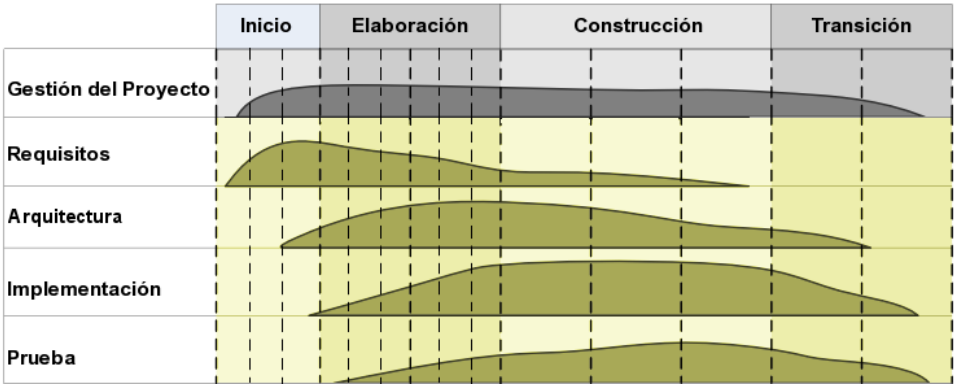


Figura 1 Ciclo de vida de OpenUP

A continuación se enumeran las principales características de OpenUP:

- Es un proceso completo, por tanto puede ser manifestado como todo el proceso para construir un sistema.
- Los procesos se pueden agregar o adaptar según lo vayan requiriendo los sistemas.
- Es un proceso ligero que proporciona una comprensión detallada del proyecto (7).

1.6 Herramienta y lenguaje de modelado

Un lenguaje de modelado es un conjunto de símbolos y de modos de disponerlos para modelar el diseño de un *software.* Durante el desarrollo del SGCLW se hará empleo del lenguaje de modelado UML[6], el cual permite elaborar los artefactos de un sistema a través de las distintas etapas de su ciclo de vida, principalmente durante el análisis y el diseño del mismo.

1.6.1 Lenguaje de Modelado Unificado

UML es un lenguaje de modelado visual que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de *software.* Organiza el proceso de diseño de tal forma que los analistas, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan de manera clara. Está compuesto por diagramas y proporciona un estándar que permite generar una vista previa del proyecto en varias facetas, de manera que sean comprensibles por el equipo, así como todo el que esté involucrado en el proceso (8). Está diseñado para utilizarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. Este lenguaje de modelado posee las siguientes características:

[6] UML (*Unified Model Language*) es un lenguaje de modelado empleado para representar el diseño de sistemas de software.

- Permite expresar de una forma gráfica un sistema de manera que pueda ser entendido por el usuario.
- Especifica cuáles son las características de un sistema antes de su construcción.
- A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Los propios elementos gráficos sirven como documentación del sistema desarrollado y pueden ser empleados para su futura revisión.

Para llevar a cabo el modelado del SGCLW se seleccionó UML en su versión 2.0. Es importante destacar que un modelo de UML describe lo que debe hacer un sistema, pero no dice cómo implementarlo.

1.6.2 Visual Paradigm for UML

Una herramienta CASE[7] es un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de un *software*, completamente o en alguna de sus fases. Favorece la calidad del sistema desarrollado, la gestión y el dominio sobre el proyecto en cuanto a su planificación, ejecución y control. Mejora la productividad y la eficacia de las áreas de desarrollo de los sistemas informáticos (9). Garantiza la consistencia de los procedimientos y verifica el uso de todos los elementos en el sistema diseñado.

Visual Paradigm for UML es una herramienta CASE que posibilita generar código y documentación para entornos integrados de desarrollo: *NetBeans*, *Eclipse* u *Oracle*. Facilita el diseño de diagramas que integran el ciclo completo de desarrollo del *software,* como el análisis, el diseño, la construcción, las pruebas y el despliegue. *Visual Paradigm* posee las siguientes características:

- Captura de requisitos mediante diagramas, modelado de casos de uso y análisis textual.
- Entorno para la especificación de detalles de casos de uso, incluyendo la especificación del modelo general y las descripciones de cada caso de uso.
- Permite la generación de código e ingeniería inversa para los lenguajes: *Java*, C, C++, PHP, XML, *Python*, C#, *Delphi* y Perl con muy alto rendimiento.

Esta herramienta permite automatizar determinados procesos. Garantiza que el modelo y el código estén actualizados, manteniendo la visión en el diseño de más alto nivel de la estructura del proyecto. Se puede aplicar en una variedad de formas para dar soporte a una metodología de desarrollo de *software,* pero no

[7]CASE (*Computer Aided Software Engineering)* se denomina a diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de un software reduciendo el costo.

especifica qué metodología o proceso utilizar. Para el diseño de la aplicación se utiliza la versión 8.0 de *Visual Paradigm for UML*.

1.7 Herramientas y lenguajes de programación

En informática, un lenguaje de programación es un conjunto de reglas definidas que describen el conjunto de acciones y servicios que un programa debe ejecutar para su funcionamiento (10). Para el desarrollo de la aplicación se seleccionaron los siguientes lenguajes de programación teniendo en cuenta sus características.

1.7.1 Lenguaje de programación PHP

PHP es un lenguaje de programación de alto nivel, interpretado y ejecutado en el servidor. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos. Permite aplicar técnicas de Programación Orientada a Objetos. Este lenguaje posee amplias y variadas fuentes de documentación, las cuales están explicadas y ejemplificadas en un único archivo de ayuda (11). Para la implementación de la aplicación se seleccionó la versión 5.4. Algunas de las características del lenguaje PHP son:

- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL**.**
- Cuenta con manejo de excepciones.
- Incremento en la velocidad y menor consumo de memoria con respecto a versiones anteriores**.**

1.7.2 Lenguaje de programación JavaScript

JavaScript es un lenguaje multiplataforma utilizado para el desarrollo de aplicaciones cliente-servidor en páginas web. Se utiliza principalmente como lenguaje de programación de lado cliente, implementado como parte de un navegador web. Permite mejoras en la interfaz de usuario y la creación de páginas *web* dinámicas (12). Este lenguaje posee como características las que a continuación se describen:

- Es dinámico, lo que significa que responde a eventos en tiempo real.
- Está orientado a objetos de forma limitada ya que no maneja los conceptos como la herencia.
- Definición literal de arreglos y objetos. Esto implica que los arreglos y objetos pueden ser creados con una sintaxis abreviada.

- Soporta expresiones regulares que proporcionan una sintaxis concisa y poderosa para la manipulación de texto, facilitando la validación de datos de tipo cadena.

Aunque existen diversos lenguajes de programación, cuando se habla de ambiente *web*, PHP y JavaScript están dentro de los más empleados. Se seleccionaron ambos lenguajes de programación debido a que cuentan con el mayor soporte para su estudio. Son fáciles de usar, sencillos de aprender y ambos trabajan en conjunto de manera eficiente.

1.7.3 Marco de trabajo

Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de programas. Incluye soporte, bibliotecas, lenguaje interpretado, entre otras herramientas para ayudar a desarrollar y unir diferentes componentes de un proyecto.

Symfony

"Es un conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones web para lenguaje PHP. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web correspondiente. Automatiza las tareas más comunes, lo que ayuda al desarrollador a dedicarse por completo a cada aspecto específico de la aplicación." (15)

Utiliza en su estructura el Modelo Vista Controlador como patrón de diseño, pues como se explicó anteriormente permite la separación de los datos y la lógica de negocio, de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Proporciona varias herramientas y clases para la disminución del tiempo de desarrollo de una aplicación *web* que sea compleja. Para el desarrollo de la aplicación se utiliza Symfony en su versión 2.4, el cual posee las siguientes características:

- Es un proyecto PHP de software libre que permite crear aplicaciones y sitios *web* mucho más rápido y seguros que las versiones anteriores debido a su alto rendimiento.
- Incluye varias herramientas gráficas y de consola para depurar fácilmente los errores que se produzcan en las aplicaciones.
- Permite establecer los parámetros de configuración de las aplicaciones a través de variables de entorno del propio servidor para evitar el uso de contraseñas en archivos de configuración.

ExtJS

Es un modelo de componentes extensibles que posee una librería JavaScript completamente orientada a objetos. Es multiplataforma y hace uso del conjunto de tecnologías para la creación de aplicaciones. Cada aplicación es enriquecida del lado del cliente. Posee gran desempeño, componentes de interfaz de usuario personalizables, con buen diseño y documentación. Para el desarrollo de la aplicación se utiliza ExtJS en su versión 4.2, este marco de trabajo posee las siguientes características:

- Brinda múltiples posibilidades para el trabajo con las validaciones, manejo de errores en el cliente y permite la personalización de temas de estilos en su utilización.
- Provee el trabajo con una amplia configuración e intenso trabajo con las hojas de estilo CSS.
- Posee una licencia de código abierto y una comercial.

Para dar solución a la problemática en cuestión se propone el empleo del marco de trabajo Symfony para gestionar el código del lado servidor. Posee una característica muy importante para el desarrollo de la solución y es el uso del patrón arquitectónico Modelo Vista Controlador, ya que al permitir la separación de estos elementos, se pueden hacer modificaciones en cualquiera de las capas sin temor a que se afecten las demás. Por otra parte, ExtJS tiene una plataforma extensa que permite la reutilización y extensión de componentes para agilizar el desarrollo de la aplicación, así como una amplia biblioteca de JavaScript, por medio de bibliotecas como JQuery y la propia nativa que ayudan a generar las interfaces de usuario del lado cliente.

1.7.4 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (IDE[8]) es un programa compuesto por un conjunto de herramientas que utilizan los programadores para desarrollar código. Las herramientas que por lo general componen un entorno de desarrollo integrado son: un editor de texto, un compilador, un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, de manera opcional, un sistema de control de versiones.

NetBeans IDE

NetBeans IDE ofrece ventajas como la creación de aplicaciones multiplataforma; proporciona facilidades y garantías para la migración, facilidad de uso, cumplimiento de regulaciones y flexibilidad entre plataformas. Dispone de soporte que permite crear y trabajar con aplicaciones *web* empleando el lenguaje

[8] IDE (*Integrated Development Environment)* es un programa informático compuesto por un conjunto de herramientas de programación.

de programación PHP. Para la implementación de la aplicación se seleccionó la versión 7.4. A continuación se enumeran características del entorno de desarrollo integrado NetBeans:

- Posee asistentes para la creación y configuración de distintos proyectos, e incluso para la elección de algunos marcos de trabajo.
- Posee un potente editor de código**,** multilenguaje, con acceso a clases mediante el código, comprobaciones sintácticas y semánticas, así como plantillas de código.
- Puede conectarse a gestores de bases de datos tales como *Oracle*, *MySQL*, *PostgreSQL* y permite autocompletar los nombres de las tablas, realizar consultas y modificaciones, todo ello integrado en el propio IDE de desarrollo.
- Se integra con servidores de aplicaciones y los gestiona desde el propio IDE.
- Es fácilmente extensible a través de *plugins*[9] (13).

Esta herramienta, al trabajar con el marco de trabajo Symfony, permite desarrollar aplicaciones de forma más sencilla y productiva. Es posible crear nuevos proyectos y aplicaciones directamente desde el IDE. Pueden ejecutarse todas las tareas de Symfony, incluso pasándole argumentos y opciones, visualizando el resultado sin necesidad de utilizar una consola de comandos externa. Además, al editar el archivo de una vista se tiene acceso al autocompletado de variables, incluso de los objetos del núcleo de Symfony.

1.7.5 Editor de texto CKEditor

"CKEditor es un editor de texto HTML que provee a la web del poder de las aplicaciones de escritorio al estilo de editores como Microsoft Word" (16). Puede ser incorporado en diferentes gestores de contenidos pues proporciona muchas de las potentes funcionalidades de editores de texto tradicionales. No requiere ningún tipo de instalación en el ordenador del usuario. Para el desarrollo de la solución se emplea la versión 3.6.2 por poseer las siguientes características:

- Posee licencia de código abierto.
- Permite la generación de código XHTML 1.0 y soporte para CSS.
- Facilita la incorporación de formularios, inserción de imágenes y la creación de tablas.
- Estrictamente personalizable y permite el formateo de fuente. Entre sus opciones está además cortar, copiar y pegar.

[9] *Plugin* (complemento en español) es una aplicación que se relaciona con otra para aportarle una nueva funcionalidad.

- Posee licencias de código abierto como GPL, LGPL y MPL que permiten la integración en *software* de uso personal y educativo, así como la integración en software comercial, teniendo en cuenta la satisfacción de los términos de las licencias.

Es importante destacar también que CKEditor, entre otras funcionalidades, permite la importación de fuente con formato previamente definido por editores de texto como Microsoft Word y Libre Office, siendo esta última la característica más importante que aporta a la solución en desarrollo. Una vez analizadas las características anteriores, se considera que con el empleo de esta herramienta es posible garantizar la gestión de contenidos de los manuales en la aplicación.

1.7.6 Servidor de aplicaciones

Un servidor es un dispositivo de software que proporciona servicios de aplicación a un conjunto de ordenadores clientes de un sistema e incluso a otros servidores. Gestiona la mayor parte o la totalidad de las funciones de lógica de negocio y de acceso a los datos de una aplicación. Los principales beneficios que oferta son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. Para atender las peticiones en función de la herramienta se seleccionó como servidor de aplicaciones el servidor web Apache.

Apache

Apache es un servidor web HTTP de código abierto que garantiza la disponibilidad en línea de la mayoría de los sitios web activos en la actualidad. El servidor está destinado a servir a una gran cantidad de plataformas web que trabajan sobre sistemas operativos como *Unix*, *Linux*, *MacOS* o sistemas operativos de *Microsoft*. La aplicación se desarrolla sobre la versión 2.4.4 por poseer las siguientes características:

- Servidor *web* flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP y HTTPS[10].
- Es multiplataforma, puede ejecutarse desde cualquier sistema operativo.
- Es modular, lo que significa que puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona y para el desarrollo de módulos específicos.
- Es extensible, por ser modular desarrolla diversas extensiones entre las que destaca el lenguaje de programación del lado del servidor PHP.

[10] HTTP y HTTPS (*Hypertext Transfer Protocol* y *Hypertext Transfer Protocol Secured*) es el protocolo usado en cada transacción de la web.

Este servidor es el adecuado para la aplicación, tomando como referencia su estructura en módulos para ser adaptado a diferentes entornos. Posee mucha diversidad en cuanto a las extensiones. Resalta su compatibilidad con el lenguaje PHP, que constituye uno de los lenguajes de programación seleccionados para el desarrollo del sistema.

1.7.7 Sistema gestor de bases de datos

Un gestor de bases de datos sirve de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Permite crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad (14). El objetivo principal de un gestor de base de datos es proporcionar un entorno práctico y eficiente, pues almacena y recupera la información necesaria para facilitar la gestión de grandes volúmenes de datos.

PostgreSQL

PostgreSQL es considerado entre los Sistemas de Base de Datos Objeto Relacional como uno de los más completos. Contiene una gran cantidad de mejoras que hacen la administración, consulta y programación en PostgreSQL mucho más fácil con respecto a otros sistemas de administración de base de datos. Esta herramienta posee las siguientes características:

- Es multiplataforma, por lo que se utiliza en sistemas operativos como *Linux* y *Windows*.
- Tiene la documentación muy bien organizada, detallada y pública.
- Excelente cumplimiento del estándar SQL.
- Altamente configurable y extensible para muchos tipos de aplicación.
- Un sofisticado analizador/optimizador de consultas.

Se determinó que el empleo de PostgreSQL es aplicable a la investigación por ser altamente escalable, tanto en la enorme cantidad de datos que puede manejar como en el número de usuarios concurrentes que soporta. Puede mantenerse en línea por períodos prolongados y requiere de poco (o en muchos casos) de ningún mantenimiento. Siguiendo línea base del departamento Desarrollo de Aplicaciones se define el empleo de PostgreSQL en su versión 9.4.

Conclusiones parciales

Partiendo de la bibliografía consultada, fueron analizados un conjunto elementos teóricos asociados al campo de acción de la investigación, lo que permitió identificar las herramientas y tecnologías adecuadas para la implementación del sistema. Se escogió como metodología a seguir OpenUP, la cual guiará todo el proceso de desarrollo del software. Para el análisis y el diseño de las funcionalidades a través de

diagramas se seleccionó *Visual Paradigm for UML* en su versión 8.0, empleando UML en su versión 2.0 como lenguaje de modelado**.** La aplicación será desarrollada utilizando el IDE NetBeans en su versión 7.4 debido a su integración con el lenguaje PHP, junto al marco de trabajo Symfony en su versión 2.4.2, empleando como lenguaje de programación PHP en su versión 5.4 para agilizar el proceso de desarrollo. Para organizar las interfaces y reutilizar componentes del lado del cliente, se emplea ExtJS en su versión 4.2, empleando el lenguaje de programación JavaScript. Se emplea la herramienta CKEditor en su versión 3.6.2, como editor de texto para garantizar la correcta gestión de los contenidos de los manuales de ayuda. Como sistema gestor de base de datos, se escoge PostgreSQL en su versión 9.4. Finalmente, se seleccionó como servidor de aplicaciones, el servidor web Apache en su versión 2.4.4.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

Introducción

En este capítulo se resume el análisis y diseño de la solución para dar respuesta a la problemática planteada. A través del modelo de dominio como representación conceptual se describe el proceso de gestión de contenidos para la generación de manuales de ayuda en DATEC. Para definir el comportamiento interno de la aplicación a desarrollar, se especifican los requisitos no funcionales y funcionales, estos últimos se agrupan en casos de usos de los que se presenta una descripción. También se identifican los patrones de diseño a utilizar en la aplicación. Este capítulo tiene como objetivo establecer la base del sistema con la arquitectura definida.

2.1 Modelo de dominio

Es la representación visual de los conceptos u objetos del mundo real en un dominio de interés. Agrupa los conceptos de un campo, siendo el mecanismo fundamental para comprender el problema y para establecer conceptos comunes (17). Un modelo del dominio muestra clases conceptuales significativas en el dominio de un problema. En la Figura 2 se muestra el modelo de dominio que describe los conceptos asociados a la gestión de contenidos para generar manuales de ayuda en el centro DATEC.

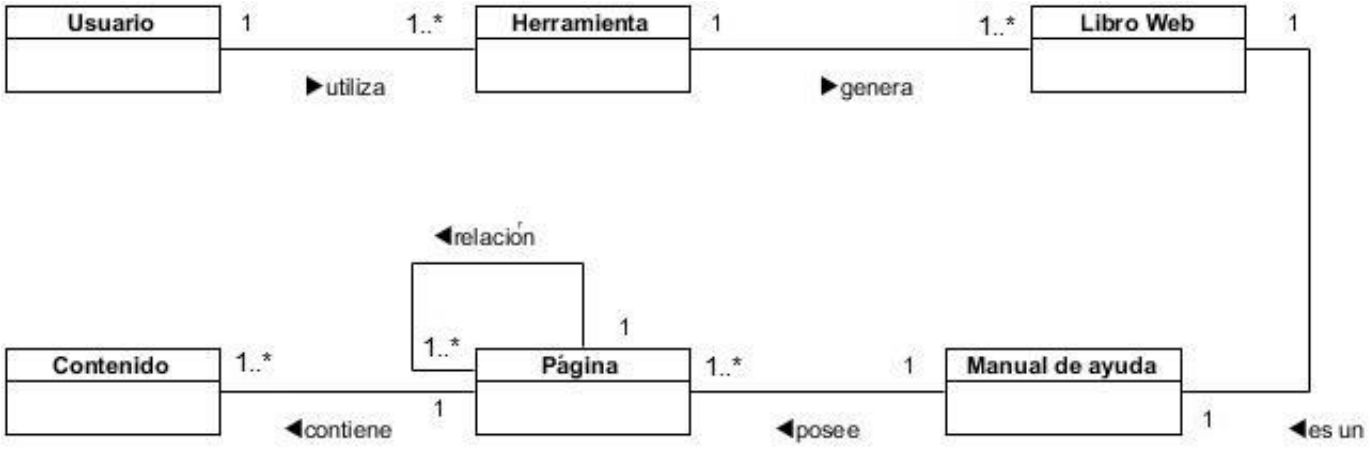


Figura 2 Modelo de dominio del SGCLW

2.1.1 Descripción de las clases del dominio

A continuación se describe cada una de las clases del modelo de dominio presentado en la Figura 2 y la relación entra estas:

- **Usuario:** refiere a la persona encargada de la elaboración de un manual de ayuda en el departamento (generalmente un analista de software).
- **Herramienta:** software empleado por el usuario para confeccionar un manual de ayuda. La herramienta empleada actualmente depende de la selección del usuario. Generalmente se emplea una herramienta de edición de texto (Microsoft Word o Libre Office) para conformar el contenido del manual de ayuda. Posteriormente un desarrollador del departamento construye un libro web manualmente haciendo empleo de las tecnologías que domina o tiene al alcance.
- **Libro web:** documento o soporte de manera general en el que se presenta un manual de ayuda. Un libro web consta de páginas relacionadas entre sí por medio de hipervínculos que conforman el árbol de contenidos del manual de ayuda.
- **Manual de ayuda:** función que desempeña el libro web dado su contenido. Generalmente incluye capturas de pantalla de diferentes escenarios de la aplicación que describe, así como pasos a seguir para llevar a cabo las funcionalidades disponibles en el sistema.
- **Página:** refiere a una sección del manual de ayuda que agrupa información relacionada a un mismo tópico. Una página puede subdividirse en otras páginas con el fin de separar su contenido en pequeñas secciones en aras de hacerlo más comprensible para el usuario. Las páginas conforman el índice o árbol de contenidos de un manual de ayuda.
- **Contenido:** representa los diferentes tipos de información (tablas, textos, imágenes e hipervínculos) que conforman una página. La variedad de estos contenidos facilitan la comprensión de un usuario sobre el contenido de un manual de ayuda, y por tanto posibilitan que un manual sea efectivo.

En el modelo de dominio un manual de ayuda es un concepto, que describe el propósito que cumple un libro web dado su contenido (ver concepto Manual de). Una vez explicadas las clases del dominio se procede a la identificación de los requisitos del sistema.

2.2 Requisitos del sistema

Los requisitos del sistema definen los servicios que proporcionan un software y las restricciones operativas de los mismos. Estos requisitos especifican el comportamiento que debe tener una aplicación

pero no aborda sobre la manera de diseñar o implementar la misma. Por esta razón, en el proceso de desarrollo de software, generalmente son empleados como punto de partida para el diseño (18).

La especificación de requisitos incluye los requisitos funcionales, que describen todas las características requeridas, lo que debe resolver el sistema; y los requisitos no funcionales o complementarios, que son aquellos que imponen restricciones en el diseño de la implementación. A continuación se presentan los requisitos funcionales del sistema.

2.2.1 Requisitos funcionales

Los requisitos funcionales (RF) son características fundamentales del sistema que expresan la capacidad de acción del mismo. Para definir un requisito funcional es necesario tener en cuenta el tipo de aplicación a desarrollar, las necesidades del cliente y los posibles usuarios finales del sistema (18). A continuación se explica detalladamente cada uno de los RF del Sistema de Gestión de Contenidos para la Generación de Libros Web. Para facilitar su comprensión, los requisitos están agrupados en diferentes tablas según las funciones que cumplen en el sistema.

Tabla 1 Requisitos funcionales asociados a la gestión de proyectos de libro web

RF#	Nombre	Descripción	Entrada/Salida
1	Adicionar PLW[11]	El sistema debe permitir al usuario crear un nuevo proyecto de libro web.	**Entrada:** datos del nuevo proyecto de libro web (título, subtítulo, autor, descripción). **Salida:** listado actualizado de proyectos de libro web.
2	Modificar PLW	El sistema debe permitir modificar un proyecto de libro web existente.	**Entrada:** valores de los atributos a modificar (título, subtítulo, autor, descripción). **Salida:** listado actualizado de PLWs.
3	Transferir PLW	El sistema debe permitir al usuario transferir un proyecto de libro web cambiándolo de esta manera de propietario.	**Entrada:** identificador (id) del proyecto de libro web a transferir y correo electrónico (email) del usuario al que se desea enviar. **Salida:** listado actualizado de proyectos de libro web.

[11] PLW es la abreviatura para Proyecto de Libro Web. Un PLW pasa a ser un Libro Web una vez que ha sido concluida su edición y es exportado del sistema.

4	Eliminar PLW	El sistema debe permitir eliminar los proyectos de libro web que están almacenados.	**Entrada:** identificador del proyecto de libro web a eliminar. **Salida:** listado actualizado de proyectos de libro web.
5	Listar PLWs	El sistema debe mostrar los proyectos de libro web que están almacenados.	**Entrada:** identificador (id) del usuario del cual se muestran los proyectos de libro web. **Salida:** listado de proyectos de libro web almacenados en el sistema.
6	Visualizar PLW	El sistema debe permitir una vista del libro web con su respectivo árbol de páginas y sus contenidos asociados.	**Entrada:** identificador del libro web a visualizar**.** **Salida:** vista del árbol de páginas del libro web.

Tabla 2 Requisitos funcionales asociados a la gestión de páginas que componen un PLW

RF#	Nombre	Descripción	Entrada/Salida
7	Adicionar Página a PLW	El sistema debe permitir al usuario crear las páginas que componen un PLW.	**Entrada:** datos de la nueva página (título, ubicación en el índice). **Salida:** árbol de páginas actualizado.
8	Modificar Página de PLW	El sistema debe permitir al usuario modificar las páginas que componen un PLW.	**Entrada:** valores de los atributos a modificar (título, ubicación en el índice). **Salida:** árbol actualizado de páginas.
9	Eliminar Página de PLW	El sistema debe permitir al usuario eliminar las páginas que componen un PLW.	**Entrada:** identificador de la página a eliminar (id). **Salida:** árbol actualizado de páginas
10	Listar Páginas de PLW	El sistema debe mostrar el árbol de páginas que componen un libro web.	**Entrada:** identificador (id) del libro web del cual se muestran las páginas. **Salida:** árbol de páginas que componen el libro web.

Tabla 3 Requisitos funcionales asociados a la edición del contenido de las páginas de un PLW

RF#	Nombre	Descripción	Entrada/Salida
11	Modificar Contenido de Página	El sistema debe permitir la modificación del contenido que se muestra en una página	**Entrada:** nuevo contenido de la página. **Salida:** contenido actualizado de la página.
12	Visualizar Contenido de Página	El sistema debe permitir al usuario obtener una vista previa del diseño de una página.	**Entrada:** identificador de la página que desea visualizar. **Salida:** vista previa de la página seleccionada.

Tabla 4 Requisitos funcionales referentes a la administración de usuarios

RF#	Nombre	Descripción	Entrada/Salida
13	Adicionar Usuario	El sistema debe crear un usuario cuando este accede por primera vez a través de una cuenta del dominio “uci.cu”, empleando el servidor LDAP.	**Entrada:** datos del nuevo usuario a insertar (usuario del dominio, nombre, apellidos, correo electrónico y rol) **Salida:** no posee salida.
14	Modificar Rol	El sistema debe permitir al administrador modificar el rol de los usuarios existentes en el sistema.	**Entrada:** valor del nuevo rol a asignar. **Salida:** listado actualizado de usuarios.
15	Eliminar Usuario	El sistema debe permitir al administrador eliminar un usuario del sistema. Al eliminar un usuario deben eliminarse todos los PLWs asociados al mismo.	**Entrada:** identificador del usuario a eliminar (id). **Salida:** listado actualizado de usuarios.
16	Listar Usuarios	El sistema debe permitir al administrador visualizar la lista de usuarios existentes en el	**Entrada:** no recibe entrada. **Salida:** listado de usuarios del sistema.

		sistema.	

Tabla 5 Requisito funcional autenticar usuario.

RF#	Nombre	Descripción	Entrada/Salida
17	Autenticar Usuario	El sistema debe permitir la autenticación de usuarios a través del servicio LDAP del dominio “uci.cu”.	**Entrada:** usuario y contraseña del dominio “uci.cu”. **Salida:** acceso al sistema o denegación del mismo.

Tabla 6 Requisitos funcionales encargados de la importación y exportación de libros web

RF#	Nombre	Descripción	Entrada/Salida
18	Exportar PLW a formato de Libro Web	El sistema debe permitir al usuario exportar un PLW, permitiendo la obtención de un manual de ayuda en formato de libro web. El libro web resultante debe estar contenido en un archivo comprimido en formato ZIP[12].	**Entrada:** identificador del PLW a exportar (id) e identificador de la plantilla en que se desea exportar un PLW. **Salida:** archivo ZIP comprimido que contiene el manual de ayuda en forma de libro web.
19	Importar Libro Web	El sistema debe permitir importar un libro web que haya sido generado por la aplicación y añadirlo como un proyecto de libro web.	**Entrada:** fichero con extensión JS que contiene los datos de un libro web generado por la aplicación. **Salida:** listado actualizado de PLWs.

Tabla 7 Requisitos funcionales asociados a la administración de plantillas de libro web

RF#	Nombre	Descripción	Entrada/Salida
20	Cargar Plantilla de Libro Web	El sistema debe permitir al administrador cargar una	**Entrada:** fichero comprimido en formato ZIP que contiene la plantilla de libro web

[12] Los archivos ZIP (.zip) son archivos individuales, algunas veces llamados "ficheros", que contienen uno o más archivos comprimidos.

		plantilla y agregarla a las plantillas de libro web disponibles.	y datos (nombre y visibilidad) de la misma. **Salida:** listado actualizado de plantillas disponibles en el sistema.
21	Eliminar Plantilla de Libro Web	El sistema debe permitir al administrador eliminar una plantilla de libro web.	**Entrada:** identificador (id) de la plantilla a eliminar. **Salida:** listado actualizado de plantillas disponibles en el sistema.
22	Modificar Plantilla de Libro Web	El sistema debe permitir al administrador modificar los atributos de las plantillas disponibles en la aplicación.	**Entrada:** nuevos valores de los atributos de la plantilla a modificar (nombre, visibilidad). **Salida:** listado actualizado de plantillas disponibles en el sistema.
23	Listar Plantillas de Libro Web	El sistema debe permitir al administrador visualizar la lista de plantillas de libro web disponibles en el sistema.	**Entrada:** no recibe entrada. **Salida:** listado de plantillas disponibles en el sistema.

2.2.2 Requisitos no funcionales

Los requisitos no funcionales (RNF) son aquellos que no se refieren directamente a las funcionalidades específicas que presenta un sistema de software, sino a las propiedades que deberá poseer o a las restricciones del mismo. El cumplimiento de estos requisitos es fundamental ya que en caso contrario el sistema puede carecer de calidad o incluso ser inutilizable (18). En las siguientes secciones se describen los requisitos no funcionales definidos para el SGCLW.

Requisitos de usabilidad

- **RNF 1 Tipo de aplicación informática**

  El SGCLW debe ser implementado haciendo empleo de tecnologías web y características visuales similares a las aplicaciones de escritorio en cuanto al diseño de las interfaces de usuario. El tipo de aplicación de software que representa el SGCLW debe ser Software basado en Web (6).

- **RNF 2 Finalidad**

El objetivo que persigue la aplicación es realizar el proceso de gestión de contenidos para la posterior generación de un libro web definiendo un formato estándar para los manuales de ayuda generados en el centro DATEC.

- **RNF 3 Software requerido para el despliegue y la utilización de la aplicación**

  Servidor de la aplicación: El servidor donde se instalará la aplicación debe contar con los siguientes requisitos de software:

  - Sistema operativo GNU/Linux (preferentemente Ubuntu en su versión 10.04).
  - El servidor debe tener instalados los siguientes paquetes: apache2, php5 y php5-pgsql.

  Servidor de bases de datos: El servidor de bases de datos de la aplicación debe contar con los siguientes requisitos de software:

  - Sistema Operativo GNU/Linux (preferentemente Ubuntu en su versión 10.04).
  - PostgreSQL en su versión 9.4.

  PC Cliente: El ordenador desde el que será empleada la herramienta debe contar con los siguientes requisitos de software.

  - Sistema operativo GNU/Linux o Windows.
  - Navegador web con soporte HTML5 (preferentemente Mozilla Firefox en su versión 28.0 o superior).

- **RNF 4 Hardware requerido para el despliegue y la utilización de la aplicación**

  PC Servidor: El servidor donde se desplegará la aplicación debe cumplir con los requisitos mínimos de hardware que se mencionan a continuación:

  - Microprocesador Pentium IV o superior, con 1.7 GHz de velocidad.
  - Memoria RAM mínimo 1Gb.
  - Disco Duro con 10Gb de capacidad para instalar el sistema.

  PC Cliente: El ordenador donde se empleará la herramienta debe cumplir con los siguientes requisitos de hardware:

  - Microprocesador Pentium IV o superior, con 1.7 GHz de velocidad.
  - Memoria RAM mínimo 512MB

Restricciones de diseño

- **RNF 5 Marco de trabajo para el desarrollo del sistema del lado del servidor**
  El sistema debe ser implementado haciendo uso del marco de trabajo Symfony en su versión 2.4, el cual propone una arquitectura tres capas haciendo empleo del patrón arquitectónico Modelo-Vista-Controlador.
- **RNF 6 Marco de trabajo para el desarrollo del sistema del lado cliente**
  Para la construcción de las interfaces visuales del sistema se propone el uso del marco de trabajo ExtJS en su versión 4.2.
- **RNF 7 Interfaces de usuario**
  La herramienta debe contar con un diseño a modo de aplicaciones RIA (*Rich Internet Application*), permitiendo al usuario una experiencia similar a la de las aplicaciones de escritorio

Requisitos para la documentación de usuarios en línea y ayuda del sistema

- **RNF 8 Manual de ayuda del sistema**
  El sistema debe contar con un manual de ayuda integrado como parte de la herramienta, de manera tal que los usuarios puedan consultar este manual para comprender los pasos a seguir con el fin de realizar las funcionalidades de la aplicación.

Requisitos de licencia

- **RNF 9 Requisitos de licencia**
  La aplicación debe ser implementada haciendo uso de herramientas y tecnologías de software libre y gratuito, cumpliendo de esta manera con las políticas de software libre de la UCI.

Requisitos de seguridad

- **RNF 10 Seguridad:**
  La seguridad del sistema se separa en los siguientes aspectos:
  - **Confidencialidad:** El sistema debe garantizar que los libros web gestionados en la aplicación estén protegidos de acceso no autorizado. Este aspecto de seguridad se garantiza por medio de la autenticación por dominio, donde un usuario tiene acceso solo a los proyectos que ha creado.
  - **Disponibilidad:** El sistema debe garantizar el acceso a la aplicación en todo momento, siempre y cuando los servidores de aplicación se encuentren activos. Los datos manejados por el sistema deben estar disponibles para el usuario al que corresponden.

2.3 Diagrama de casos de uso de sistema

El diagrama de casos de uso de sistema muestra la interacción entre los actores y las funcionalidades de un sistema de software agrupadas en casos de uso. Los casos de uso (CU) agrupan un conjunto de requisitos funcionales que indican qué hará el sistema, mientras que los actores no son más que individuos que interactúan con estos.

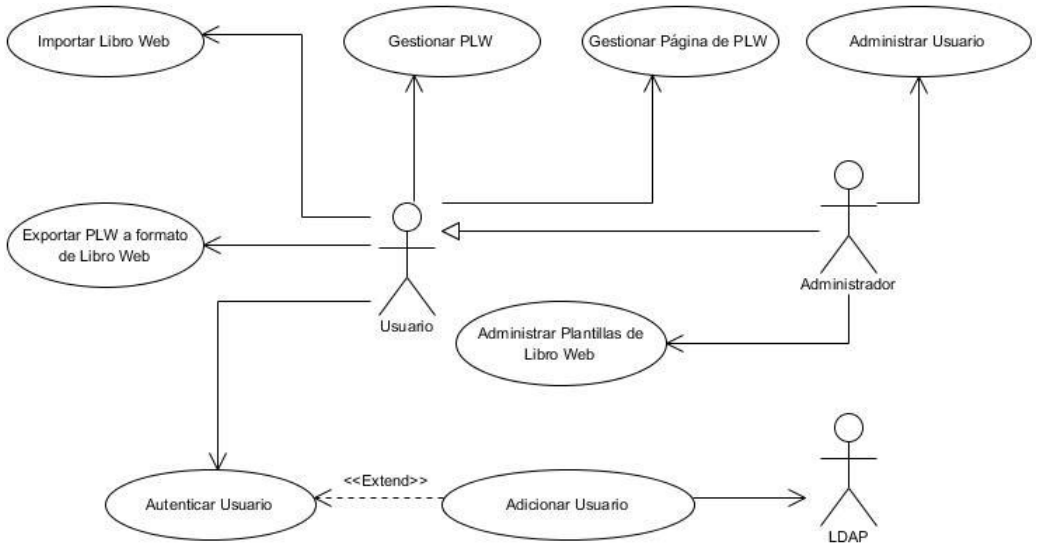


Figura 3 Diagrama de casos de uso del SGCLW

En la Figura 3 se muestra el diagrama de casos de uso perteneciente al SGCLW. Para la confección del mismo se hizo empleo de los patrones de casos de uso que se describen en la siguiente sección.

2.3.1 Patrones de casos de uso

Los patrones de casos de uso tienen como objetivo representar las funcionalidades del sistema de manera tal que el modelo sea reutilizable y entendible. Para el desarrollo del SGCLW se hizo empleo de los patrones de CU que se enumeran a continuación:

- **CRUD:** Se utiliza cuando las operaciones pueden ser realizadas sobre una parte de la información de un tipo específico (parcial) o sobre la totalidad de la información de dicho tipo (total). En el CU Gestionar PLW evidencia el empleo del patrón CRUD total, en él se agrupan requisitos para crear, modificar, obtener y eliminar los proyectos de libro web, siendo de la misma manera para el CU Gestionar Página de PLW. En el caso del CU Administrar Usuario se evidencia el uso del patrón CRUD parcial, ya que la acción correspondiente a la creación de usuarios sucede fuera del CU.

- **Múltiples Actores con Rol Común:** Este patrón indica que dos actores juegan el mismo rol sobre un caso de uso. Evidenciado en la relación de herencia entre Usuario y Administrador.
- **Extensión Concreta:** Este patrón se aplica cuando un flujo puede extender del flujo de otro caso de uso. Evidenciado en la relación entre el CU Autenticar Usuario y el CU Adicionar Usuario, donde la autenticación de un usuario a través del servicio LDAP puede desencadenar la creación de un nuevo usuario del sistema.

2.3.2 Descripción de los actores

En la Figura 3 Diagrama de casos de uso del se representan tres actores: Usuario, Administrador y LDAP. El actor Usuario refiere a la persona que interactúa con el sistema con el fin de crear y posteriormente generar un manual de ayuda en forma de libro web. El actor Administrador es una especialización del actor Usuario, por lo que puede desarrollar las mismas tareas; además de llevar a cabo la administración de los usuarios y las plantillas de libro web del sistema. Por otra parte, el actor LDAP representa al sistema de autenticación del dominio “uci.cu”, el cual provee los datos de los usuarios y valida la autenticación de los mismos en el SGCLW. A continuación se describe cada uno de los casos de uso existentes en el diagrama.

2.3.3 Descripción general de los casos de uso

CU Gestionar PLW

Gestionar PLW constituye el CU arquitectónicamente significativo de la herramienta. El mismo agrupa las funcionalidades que permiten la creación y edición de un proyecto de libro web. A continuación se enumeran los requisitos funcionales que satisface este caso de uso: RF1, RF2, RF3, RF4, RF5 y RF6.

CU Gestionar Página de PLW

El caso de uso Gestionar Página de PLW agrupa las funcionalidades que permiten el trabajo con las páginas que componen el árbol de contenidos de un libro web. El mismo agrupa los siguientes requisitos funcionales: RF7, RF8, RF9, RF10, RF11 y RF12.

CU Adicionar Usuario

El caso de uso Adicionar Usuario contiene la funcionalidad que permite agregar un usuario al sistema. Esta funcionalidad se ejecuta de manera automática. Una vez confirmadas las credenciales de un usuario, se verifica si este posee una entrada en la tabla de usuarios del sistema, de no ser así es añadido. El requisito funcional contenido en este CU es el RF13.

CU Administrar Usuario

El caso de uso Administrar Usuario es inicializado por el actor Administrador y satisface las funcionalidades que permiten el trabajo con los usuarios del sistema. El mismo agrupa los siguientes requisitos funcionales: RF14, RF15 y RF16.

CU Autenticar Usuario

Este CU está asociado al RF17 y es el encargado de inicializar el CU Adicionar Usuario. La autenticación en el sistema se realiza través del dominio “uci.cu” empleando el servicio de autenticación LDAP.

CU Exportar PLW a formato de Libro Web

Este CU está asociado al RF18 y permite al usuario la obtención de un manual de ayuda en forma de libro web a partir de un PLW que se encuentra almacenado en el SGCLW. El libro web resultante es un archivo comprimido en formato ZIP.

CU Importar Libro Web

Este CU permite al usuario seleccionar desde su ordenador un fichero JS que contiene los datos de un PLW exportado por el SGCLW. Este fichero se encuentra dentro del archivo comprimido en formato ZIP resultante de la exportación de cada libro web. El CU Importar Libro Web satisface el RF 19.

CU Administrar Plantillas de Libro Web

El caso de uso Administrar Plantillas de Libro Web es inicializado por el actor Administrador y engloba las funcionalidades que permiten el trabajo con las plantillas en que se exportan los libros web del sistema. El mismo engloba los siguientes requisitos funcionales: RF20, RF21, RF22 y RF23.

2.3.4 Descripción detallada del caso de uso significativo del sistema

Tabla 8 Descripción del CU Gestionar PLW

Objetivo	Llevar a cabo la gestión de los proyectos de libro web en el sistema.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción “Mis Proyectos” ubicada en la interfaz principal de la aplicación. Seguidamente selecciona una de las siguientes opciones de la interfaz “Mis proyectos”: • Adicionar • Editar • Eliminar • Contenido • Transferir
Complejidad	Alta
Prioridad	Crítico
Referencias	RF1, RF2, RF3, RF4, RF5, RF6
Precondiciones	• El usuario debe estar autenticado en el sistema. • Para llevar a cabo las funcionalidades Editar, Eliminar, Contenido y Transferir el usuario debe tener creado al menos un PLW.
Pos-condiciones	En dependencia de la opción seleccionada: • Se crea un nuevo PLW. • Se modifica un PLW existente. • Se elimina un PLW del sistema. • Se visualizan los contenidos actuales del PLW seleccionado. • Se cambia el propietario de un PLW seleccionado.
Flujo de eventos	
Flujo básico del CU Gestionar PLW	

Actor	**Sistema**
1. Selecciona ¨Mis proyectos¨ en la interfaz principal de la aplicación.	2. Muestra la interfaz ¨Mis proyectos¨ con un listado de los proyectos de libro web asociados al usuario autenticado y el conjunto de botones que permiten las

	operaciones de gestión de proyectos de libro web.
3. Selecciona la operación que desea realizar a través de la barra de herramientas de la interfaz o del menú contextual desplegado con el clic secundario sobre la lista de proyectos. a. Adicionar b. Editar c. Eliminar d. Contenido e. Transferir	4. Permite realizar operaciones de gestión de proyectos de libro web en dependencia de la operación solicitada: a. Adicionar un PLW. (Ver sección 1: Adicionar PLW) b. Editar un proyecto. (Ver sección 2: Modificar PLW) c. Eliminar un proyecto. (Ver sección 3: Eliminar PLW) d. Visualizar el contenido de un proyecto de libro web. (Ver sección 4: Visualizar PLW) e. Cambiar el propietario de un libro web. (Ver sección 5: Transferir PLW) Termina el caso de uso

Prototipo de la interfaz

Figura 4 Interfaz Mis Proyectos

Sección 1: "Adicionar PLW"

Flujo básico Adicionar PLW

Actor	Sistema
	1. Muestra la interfaz ¨Propiedades del proyecto¨ con los campos título, subtítulo, autor y descripción.

2. Rellena los campos con las propiedades del proyecto de libro web que desea crear. Selecciona la opción ¨Aceptar¨ para indicar que se cree el proyecto de libro web.	3. Verifica que no existan campos vacíos. En caso de que los datos no estén completos ver Flujo alterno 1 ¨Campos vacíos¨. Si los campos se introdujeron correctamente, registra el nuevo proyecto de libro web y regresa al Flujo básico del CU Gestionar PLW.

Prototipo de la interfaz

Figura 5 Interfaz Propiedades del proyecto

Flujos alternos

Flujo alterno 1: Campos vacíos <El usuario no completó todos los campos requeridos>

Actor	Sistema
	1. Muestra los bordes de los campos que deben ser llenados de color rojo.
2. Completa los campos vacíos.	3. Procede con el Flujo básico de la sección 1 ¨Adicionar PLW¨ Paso 3.

Flujo alterno 2: Cancelar <El usuario cancela la operación>

Actor	Sistema
1. Selecciona la opción ¨Cancelar¨ para abortar la operación.	2. Regresa al Flujo básico del CU Gestionar PLW.

Sección 2: "Modificar PLW"

Flujo básico Modificar PLW

Actor	Sistema
	1. Muestra la interfaz ¨Propiedades del proyecto¨ con los

	campos título, subtítulo, autor y descripción.
2. Modifica los campos con las propiedades del proyecto de libro web que desea crear. Seguidamente selecciona la opción ¨Aceptar¨ para indicar que se modifique el proyecto de libro web.	3. Verifica que no existan campos vacíos. En caso de que los datos no estén completos ver Flujo alterno 1 ¨Campos vacíos¨. Si los campos se introdujeron correctamente actualiza los valores del proyecto de libro web. Seguidamente regresa al Flujo básico del CU Gestionar PLW.
Flujos alternos	
Flujo alterno 1 Campos vacíos <El usuario no completó todos los campos requeridos>	
Actor	**Sistema**
	1. Muestra los bordes de los campos que deben ser llenados en color rojo.
2. Completa los campos vacíos.	3. Procede con el Flujo básico de la sección 1 ¨ Adicionar PLW¨ Paso 3.
Flujo alterno 2: Cancelar <El usuario cancela la operación>	
Actor	**Sistema**
1. Selecciona la opción ¨Cancelar¨ para abortar la operación.	2. Regresa al Flujo básico del CU Gestionar PLW.
Sección 3: “Eliminar PLW”	
Flujo básico Eliminar PLW	
Actor	**Sistema**
	1. Muestra un mensaje de diálogo pidiendo confirmación para la operación de eliminar el PLW.
2. Selecciona una opción del mensaje de diálogo: a. Sí b. No	3. Realiza una operación en dependencia de la opción seleccionada: a. Elimina el proyecto de libro web del sistema b. No elimina el proyecto del sistema Seguidamente regresa al flujo básico del CU Gestionar PLW.
Prototipo de la interfaz	

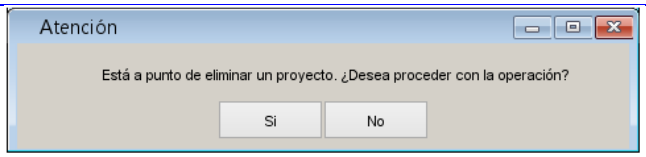


Figura 6 Mensaje de diálogo Eliminar proyecto

Flujos Alternos	
Flujo alterno 1: Cancelar <El usuario cancela la operación>	
Actor	**Sistema**
1. Selecciona la opción ¨Cancelar¨ para abortar la operación.	2. Regresa al Flujo básico del CU Gestionar PLW.
Sección 4: "Visualizar PLW"	
Flujo básico Visualizar PLW	
Actor	**Sistema**
	1. Muestra la interfaz ¨Contenido¨ la cual contiene el índice de páginas del PLW a través del cual se puede navegar y editar sus contenidos. Ver descripción del CU Gestionar Página de PLW.
Sección 5: "Transferir PLW"	
Flujo básico Transferir PLW	
Actor	**Sistema**
	1. Muestra la interfaz ¨Transferir Proyecto¨ con un campo que permite elegir el correo electrónico de uno de los usuarios del sistema al que se le transferirá el proyecto de libro web.
2. Selecciona uno de los usuarios existentes en el sistema y elige la opción ¨Aceptar¨.	3. Cambia el propietario del proyecto de libro web al usuario seleccionado y regresa al flujo básico del CU Gestionar PLW.
Prototipo de la interfaz	

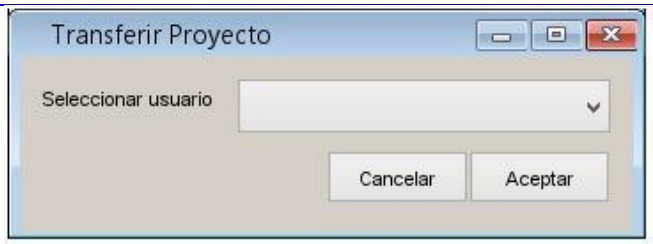


Figura 7 Interfaz Transferir proyecto

Flujos Alternos		
Flujo alterno 1: Cancelar <El usuario cancela la operación>		
Actor		**Sistema**
3. Selecciona la opción ¨Cancelar¨ para abortar la operación.		4. Regresa al Flujo básico del CU Gestionar PLW.
Relaciones	**CU Incluidos**	No procede
	CU Extendidos	No procede
Requisitos no funcionales	**RNF 1** Tipo de aplicación informática **RNF 6** Marco de trabajo para el desarrollo del sistema del lado cliente **RNF 7** Interfaces de usuario **RNF 10** Seguridad	
Asuntos pendientes	No procede.	

2.4 Modelo de diseño

El modelo de diseño se puede considerar como una abstracción del modelo de implementación de un sistema y se utiliza como una entrada fundamental en este proceso. Su objetivo es especificar una solución gráfica que pueda ser convertida posteriormente en código fuente. Para cumplir su objetivo se puede emplear el lenguaje de modelado UML para confeccionar diagramas de paquetes y de clases.

2.4.1 Diagrama de paquetes

La organización de elementos en paquetes ofrece la ventaja de separar los elementos detallados en abstracciones más amplias, lo cual brinda soporte a una vista de nivel superior y permite contemplar el modelo en agrupamientos más simples (18). Este tipo de diagrama se emplea para reflejar la organización de los paquetes presentes en la aplicación y sus respectivos elementos. Symfony 2.4 propone una estructura predefinida de directorios para organizar el código fuente de la aplicación.

El directorio *Controller* contendrá los controladores del SGCLW. Estos son los responsables de la gestión de los usuarios de la aplicación, la autenticación a través del servicio LDAP y la gestión de los libros web.

Para cumplir con esta responsabilidad utiliza las entidades que se encuentran en el directorio *Entity*, los repositorios almacenados en *Repository* y los recursos de la aplicación del directorio *Resources*. Las entidades son las clases encargadas de la abstracción de la lógica relacionada con los datos, proporcionando el uso de los mismos con independencia del gestor de bases de datos utilizado. El directorio *Repository* contiene los archivos que contienen las sentencias del lenguaje DQL (*Doctrine Query Language*), propio de Doctrine[13]. El directorio *Dependency Injection* en Symfony 2.4 por defecto contiene los archivos necesarios para el correcto funcionamiento del Inyector de Dependencias[14].

Por otra parte, el directorio *Resources* contendrá las carpetas *view, translation, public y config*. Contenidas en *view* se encontrarán las plantillas *Twig*[15] encargadas de mostrar los componentes del sistema. La carpeta *public* contiene todos los ficheros y clases pertenecientes al lado cliente, en este directorio se encuentran el marco de trabajo ExtJS y el editor de texto CKEditor. En la Figura 8 se muestra el diagrama de paquetes del SGCLW.

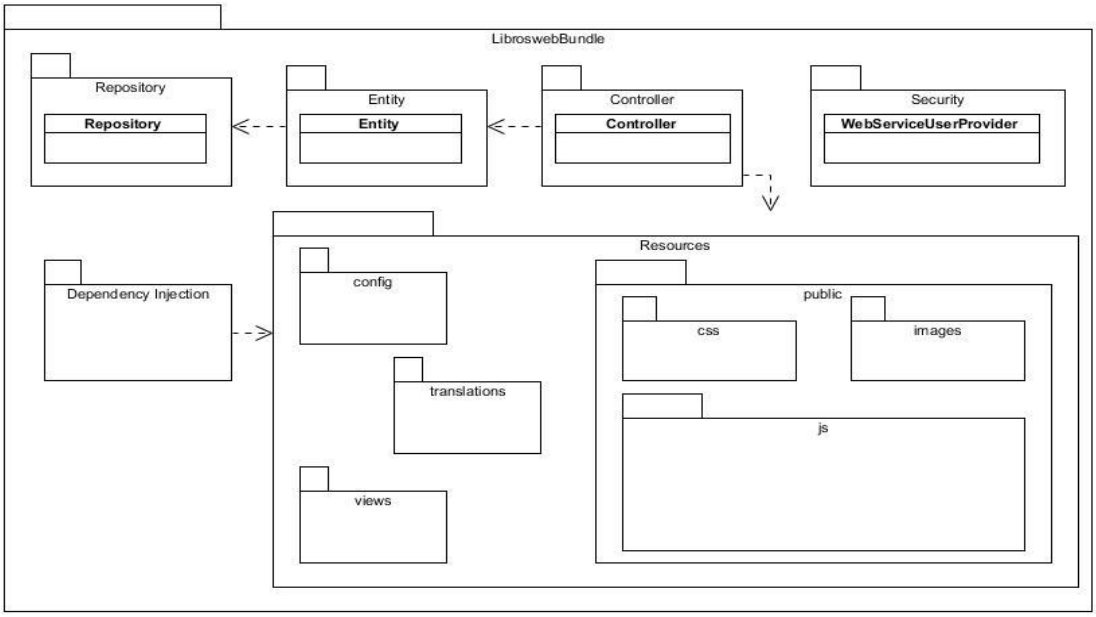


Figura 8 Diagrama de paquetes del SGCLW

[13] Doctrine es un ORM (Object Relational Mapper) realizado en PHP, que es una capa de abstracción de la base de datos centrada en el mapeo de objetos.
[14] Es un componente que permite estandarizar y centralizar la forma en que se construyen los objetos en una aplicación.
[15] Twig es un motor de plantillas para PHP cuyo objetivo según sus creadores es ofrecer una alternativa flexible, potente y segura a otros motores similares.

2.4.2 Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Generalmente contiene las clases, asociaciones, atributos, interfaces, métodos y dependencias (18). A continuación en la Figura 9 se presenta el diagrama de clases del diseño correspondiente al CU Gestionar Libro Web.

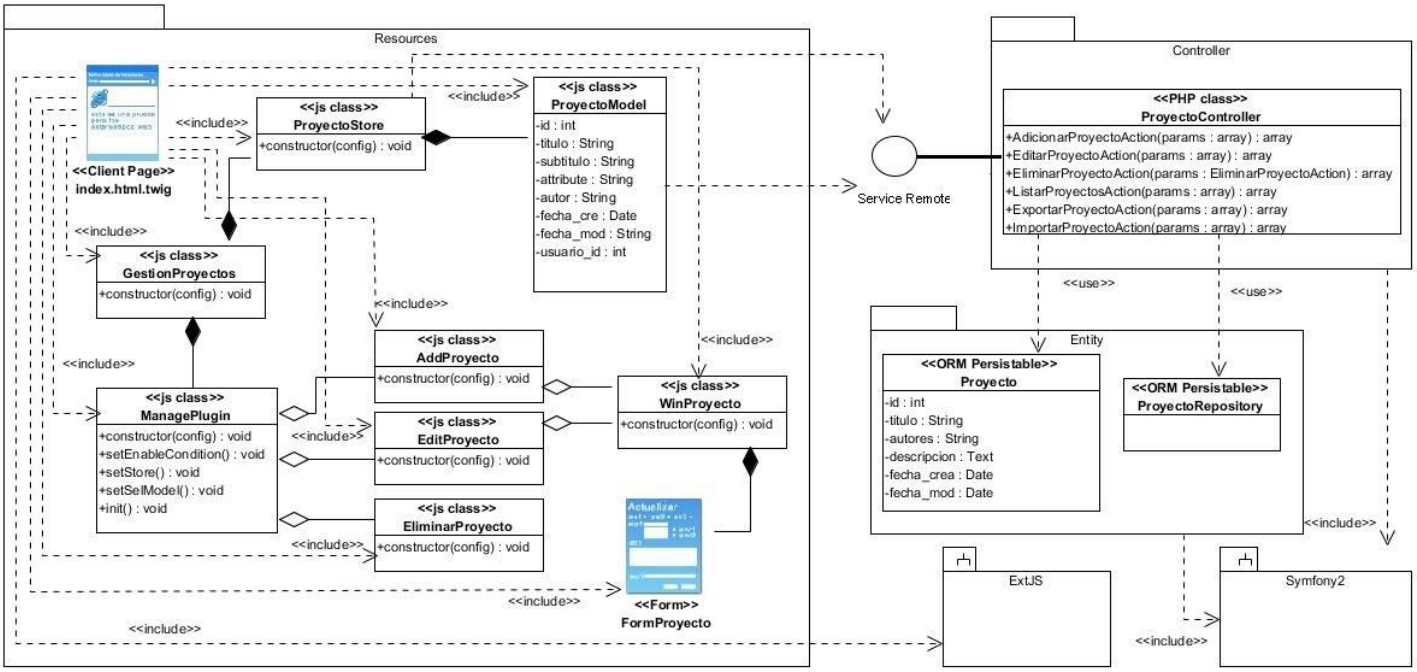


Figura 9 Diagrama de clases del diseño correspondiente al caso de uso Gestionar PLW

El diagrama de clases representado anteriormente contiene las principales clases que intervienen en el proceso de gestión de libros web con sus principales métodos, atributos y relaciones. Para lograr un mejor entendimiento del mismo se han agrupado las clases pertenecientes al dominio en los paquetes *Entity*, *Resources* y *Controller*. A continuación se explica cada una de las clases del dominio.

El paquete *Entity* contiene las clases encargadas del acceso a datos, estas clases están definidas según la estructura que propone Symfony2. Este paquete contiene la clase Proyecto que representa la entidad que contiene los atributos principales de un libro web, así como los métodos para acceder y modificar estos atributos. Por otra parte, la clase ProyectoRepository contiene los métodos que permiten realizar búsquedas sobre el conjunto de libros web existentes en el sistema.

Contenidas en el paquete *Resources* se encuentran las clases encargadas de presentar la información al usuario. El fichero index.html.twig es una página cliente la cual contiene todas las clases asociadas a la

vista, a través de esta página se presenta el sistema al usuario. Las clases ProyectoStore y ProyectoModel son las encargadas de almacenar de manera temporal las entidades de tipo libro web que son manejadas por la vista. La clase ManagePlugin contenida dentro de la clase GestionProyectos maneja los eventos resultantes de la interacción con el usuario, controlando el estado de los formularios y botones de la aplicación. La creación, edición y eliminación de libros web en el sistema se lleva a cabo en las clases AddProyecto, EditProyecto y EliminarProyecto respectivamente, empleando la clase WinProyecto que posee el formulario con los datos de un libro web.

El paquete *Controller* posee la clase PHP ProyectoController que contiene los métodos solicitados por la vista para realizar acciones sobre el modelo. Este paquete se corresponde con la capa controladora del patrón arquitectónico Modelo Vista Controlador (MVC) descrito en la siguiente sección.

2.4.3 Patrón arquitectónico MVC

Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que se compone de subsistemas, responsabilidades e interrelaciones (18). Symfony implementa en su estructura el patrón Modelo Vista Controlador, por tanto es el patrón de arquitectura utilizado durante el desarrollo de la aplicación.

Modelo Vista Controlador (MVC)

El patrón MVC es una variante de la arquitectura por capas que permite realizar la programación multicapa, separando en tres componentes distintos los datos de una aplicación, la interfaz del usuario y la lógica de control. A continuación se explican las capas presentes en este patrón.

El Modelo representa la información con la que trabaja el sistema. Gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Las peticiones de acceso o manipulación de la información llegan al modelo a través del controlador. La capa perteneciente al modelo según el diagrama de clases del diseño representado en la Figura 9 se corresponde con el paquete *Entity*.

La Vista presenta al modelo en un formato adecuado para interactuar con el usuario, se puede ver reflejado en la interfaz de usuario, por tanto requiere del modelo para obtener la información que debe representar como salida. Además, posee los formularios que recopilan la información para introducir nuevos datos al propio modelo. El paquete *Resources* presente en la Figura 9 se corresponde con la vista según propone el patrón arquitectónico MVC.

El Controlador responde a eventos que son usualmente acciones del usuario, e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información. Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente. En el diagrama representado en la Figura 9, el controlador se corresponde con el paquete *Controller* definido por el marco de trabajo Symfony2.

El Modelo Vista Controlador utiliza el concepto de capas, pero de una manera personalizada para realizar su función. Cada componente es independiente del otro sin dejar de interactuar entre ellos, lo que permite mayor flexibilidad y organización de la aplicación que se desarrolla.

2.5 Patrones de diseño identificados en la solución

"Los patrones ayudan a construir sobre la experiencia colectiva de ingenieros de software experimentados. Estos capturan la experiencia existente y que ha demostrado ser exitosa en el desarrollo de software, y ayudan a promover las buenas prácticas de diseño. Cada patrón aborda un problema específico y recurrente en el diseño o implementación de un software" (19). En la actualidad su uso en el desarrollo de aplicaciones informáticas juega un papel fundamental. El empleo de patrones permite reducir el tiempo de desarrollo de un software y aporta a la calidad del producto final. El presente epígrafe describe los patrones de diseño empleados durante el desarrollo de la investigación.

2.5.1 Patrones GRASP[16]

Los Patrones Generales de *Software* para la Asignación de Responsabilidades (GRASP), proponen los principios fundamentales de diseño de objetos para la asignación de responsabilidades y aplican el razonamiento para el diseño de una forma sistemática, racional y explicable (20). A continuación se describen los patrones GRASP identificados el SGCLW:

- **Experto**: En el caso de uso crítico (CU Gestionar PLW), la clase experta en información es la entidad Proyecto perteneciente al paquete Modelo (véase Figura 9 Diagrama de clases del diseño correspondiente al caso de uso Gestionar PLW)**.** Esta clase contiene los métodos que permiten la creación, modificación y acceso a los proyectos, y por lo tanto es la clase responsable del trabajo con las entidades de tipo proyecto.
- **Creador**: Este patrón tiene como propósito encontrar un creador que tenga que conectarse con el objeto producido en cualquier evento. En la Figura 10 se muestra un fragmento del diagrama de clases

[16]GRASP (*General Responsibility Assignment Software Patterns)* son patrones generales de software para asignación de responsabilidades.

del diseño del CU Gestionar PLW, donde se evidencia el empleo de este patrón. La clase controladora ProyectoController es la responsable de crear instancias de entidades de tipo proyecto.

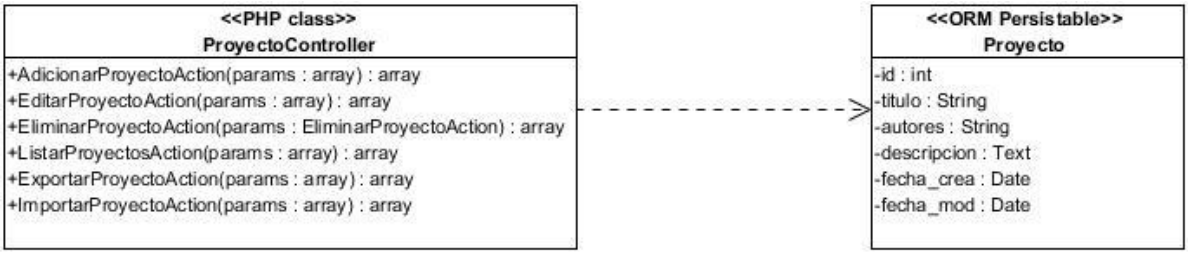


Figura 10 Aplicación del patrón Creador

- **Alta Cohesión**: Este patrón se aplica en la solución para obtener clases que almacenen información a fin a su propósito y no se vean sobrecargadas con funcionalidades que no le correspondan lógicamente. Un ejemplo de aplicación del patrón alta cohesión se puede observar en la Figura 10 entre las clases ProyectoController y Proyecto. La clase controladora ProyectoController posee los métodos encargados de la gestión de proyectos de manera general, pero para llevar a cabo la inserción hace empleo de la clase entidad Proyecto, que posee los métodos para gestionar cada uno de los atributos de un proyecto en la base de datos.
- **Bajo Acoplamiento:** Este patrón se aplica en la solución para lograr una dependencia escasa con otras clases y promover el aumento de la reutilización de las mismas. En la Figura 11 se puede apreciar un ejemplo de reutilización de las clases gracias a la aplicación del patrón bajo acoplamiento entre las clases de la vista.

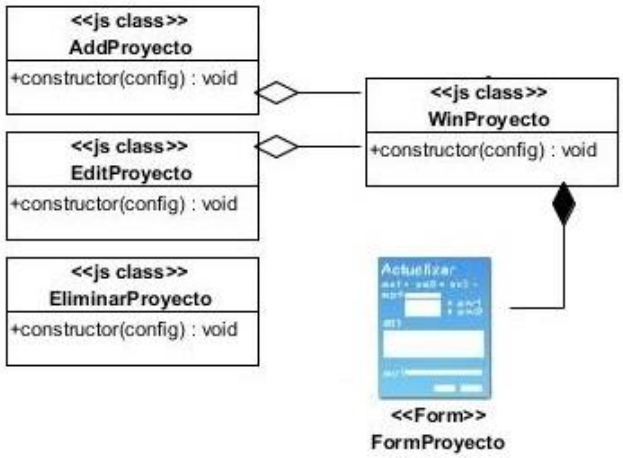


Figura 11 Ejemplo de reutilización de las clases dentro del SGCLW

Las clases AddProyecto, EditProyecto y EliminarProyecto se encargan de controlar los eventos correspondientes a la adición, edición y eliminación de los proyectos respectivamente. Por otra parte la clase WinProyecto posee el formulario con los campos pertenecientes a un proyecto, de manera que puede ser reutilizada en las clases encargadas de la adición y edición de proyectos para mostrar dicho formulario.

- **Controlador**: Este patrón sugiere que la lógica de negocio esté separada de la capa de presentación. La estructura que propone Symfony garantiza el empleo de este patrón, definiendo clases controladoras para cada entidad del sistema. Cada una de estas clases posee responsabilidades específicas de controlar el flujo de eventos del sistema. En la Figura 10 aparece la clase ProyectoController que posee los métodos necesarios para manejar el flujo de eventos asociados a la gestión de libros web.

2.5.2 Patrones GOF[17]

Los patrones *GOF* fueron sugeridos en 1995 por la banda de los cuatro Erich Gamma, Richard Helm, Ralph Johnson y John Vissidess. Estos promueven un avance en la programación orientada a objetos ya que ayudan a clasificarla según su propósito en tres clases como se enumera a continuación:

- Los **Patrones creacionales** crean instancias de clases y definen el uso que recibirán estas una vez creadas.
- Los **Patrones estructurales** definen la relación entre las clases dando lugar a la combinación y formación de estructuras mayores.
- Los **Patrones de comportamiento** engloban aquellos patrones de software que se centran en la comunicación entre las distintas clases y objetos.

El empleo de los marcos de trabajo Symfony y ExtJS define el uso de patrones GOF, potenciando la reutilización de los componentes del sistema. En el fragmento del diagrama de clases del diseño que aparece en la Figura 12 se puede observar el empleo del **patrón fachada** perteneciente a los patrones estructurales, la clase index.html.twig proporciona una interfaz unificada para el conjunto de interfaces de la vista. La clase ManagePlugins implementa el **patrón observador** perteneciente al grupo de patrones de comportamiento, esta clase posee los objetos que garantizan la creación, edición y eliminación de los

[17] GOF *(Gang of Four)* representan una base para la búsqueda de soluciones a problemas comunes en el desarrollo de un software.

proyectos, de forma tal que cuando sucede un cambio de estado en uno de ellos, se notifica y actualizan automáticamente todos los objetos.

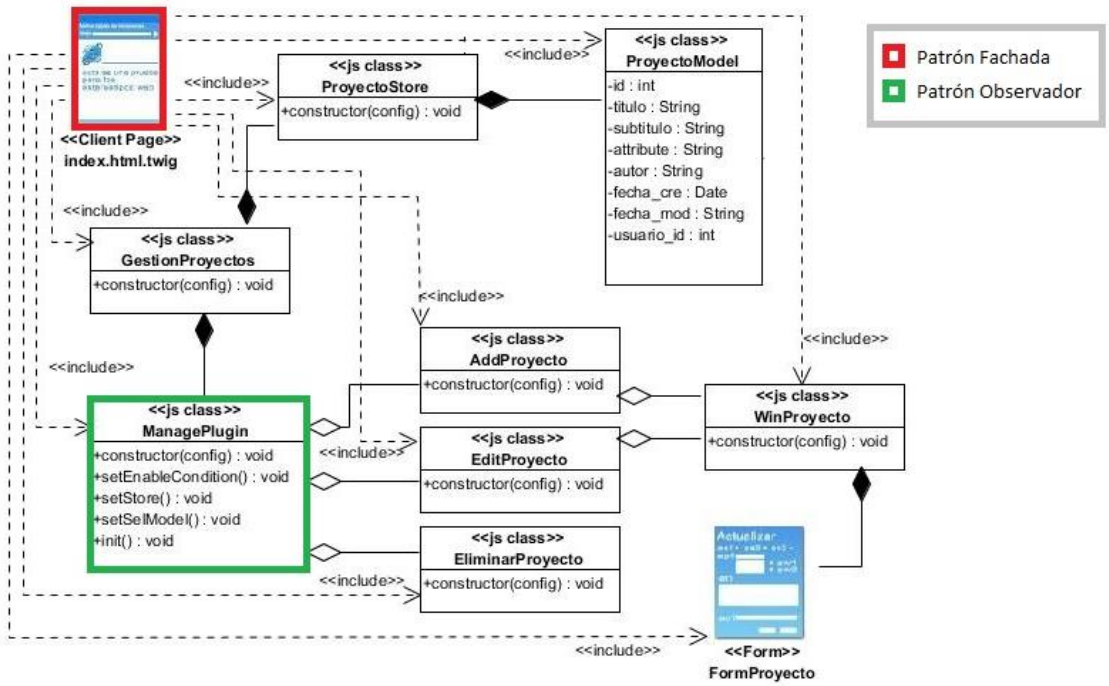


Figura 12 Aplicación de patrones GOF

2.6 Modelo de datos

El modelado de datos persigue el objetivo de identificar y representar gráficamente las entidades que participan en el sistema. En él se especifican las relaciones entre las entidades y sus atributos para brindar una información más completa al equipo de desarrollo. En la Figura 13 se muestra el modelo de datos del SGCLW.

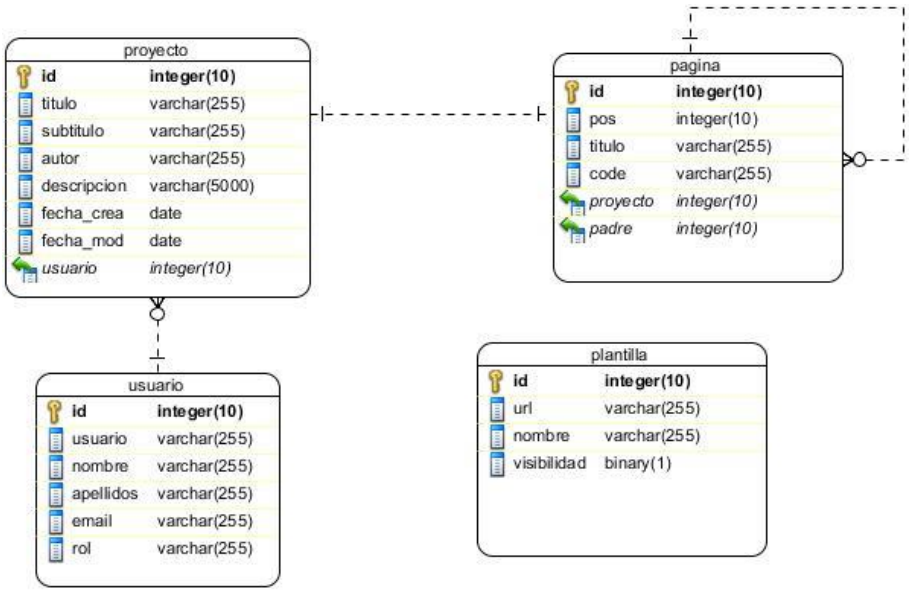


Figura 13 Modelo de Datos del SGCLW

2.6.1 Descripción del modelo de datos

El modelo de datos cuenta con cuatro entidades: usuario, proyecto, página y plantilla. La entidad usuario contiene los atributos de los usuarios que interactúan con la aplicación, esta entidad no almacena la clave de acceso del usuario puesto que la autenticación se lleva a cabo a través del servicio de autenticación LDAP del dominio uci.cu. Proyecto es la entidad que almacena los proyectos de libro web que finalmente serán exportados por la aplicación, como indica su relación con la entidad usuario: un usuario puede tener asociado tantos proyectos como desee. Lógicamente un proyecto de libro web está compuesto por múltiples páginas de contenidos, es por esto que la entidad proyecto posee una relación de uno a muchos con la entidad página. La entidad página se relaciona a si misma de manera recursiva con el fin de representar el árbol de contenidos del libro web, donde cada página puede subdividir sus contenidos en otras páginas con el fin de conformar un manual de ayuda comprensible para los usuarios. La entidad plantilla se encarga de almacenar la información de las plantillas en que pueden ser exportados los libros web del sistema. Las plantillas definen el aspecto que finalmente tendrá un libro web al ser exportado de la aplicación. Para definir de manera detallada los atributos de las entidades usuario, proyecto, página y plantilla se presentan a continuación en las tablas: Tabla 9, Tabla 10, Tabla 11 y respectivamente.

Tabla 9 Descripción de la entidad usuario

Usuario		
Atributo	Tipo de dato	Descripción
id	Integer	Identificador único en el sistema.
user	Varchar	Identificador del usuario en el dominio.
nombre	Varchar	Nombre del usuario.
apellidos	Varchar	Apellidos del usuario.
email	Varchar	Correo electrónico del usuario.
rol	Varchar	Rol que juega el usuario en el sistema.

Tabla 10 Descripción de la entidad proyecto

proyecto		
Atributo	Tipo de dato	Descripción
id	Integer	Identificador único en el sistema.
titulo	Varchar	Título del libro web.
subtitulo	Varchar	Subtítulo del libro web.
autor	Varchar	Nombre del autor del libro web.
descripcion	Text	Breve descripción del libro web.
fecha_crea	date	Fecha en que se creó el libro web.
fecha_mod	date	Fecha en que se modificó por última vez el libro web.

Tabla 11 Descripción de la entidad página

página		
Atributo	Tipo de dato	Descripción
id	integer	Identificador único en el sistema.
pos	integer	Posición de la página en el árbol de contenidos del libro.
titulo	varchar	Título con que se presenta la página en el árbol de contenidos del libro.
code	varchar	Código HTML que conforma el contenido de la página.

Tabla 12 Descripción de la entidad plantilla

plantilla		
Atributo	Tipo de dato	Descripción
id	integer	Identificador único en el sistema.
url	varchar	Dirección en el servidor donde se encuentra el archivo que contiene la plantilla de libro web.
nombre	varchar	Nombre con que se presenta la plantilla al usuario.
visibilidad	binary	Visibilidad de la plantilla: “1” si la plantilla es pública y puede ser usada por todos los usuarios, “0” si la plantilla solo puede ser empleada por analistas.

2.7 Modelo de despliegue

Los diagramas de despliegue muestran las relaciones físicas entre los componentes físicos y lógicos del sistema final. Representa los artefactos del sistema como nodos, los cuales son conectados a través de caminos de comunicación para crear redes de sistemas de complejidad arbitraria (17). En la Figura 14 se muestra el diagrama de despliegue del SGCLW.

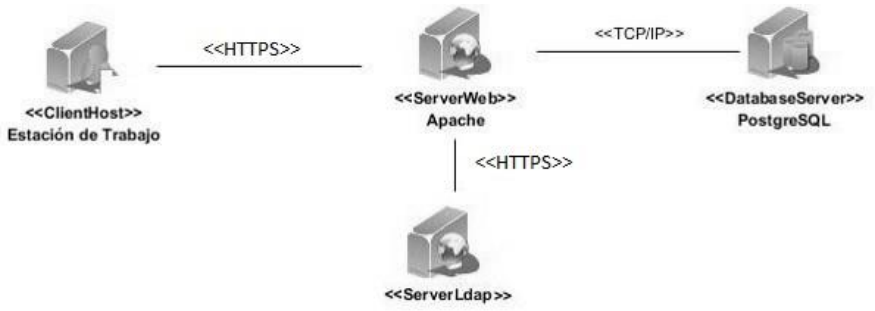


Figura 14 Diagrama de despliegue del sistema.

El modelo de despliegue cuenta con cuatro nodos. El nodo Estación de Trabajo representa el ordenador donde será ejecutada la aplicación, el cual requiere una conexión HTTPS con el servidor web ubicado en el nodo Apache. El servidor web realiza operaciones sobre el servidor de base de datos representado por el nodo PostgreSQL a través del protocolo TCP/IP, este servidor además requiere de una conexión HTTPS al servidor de autenticación de dominio LDAP para comprobar las credenciales de los usuarios del sistema.

Conclusiones parciales

Las tareas de la investigación asociadas al análisis y diseño permitieron la captura de los requisitos del sistema, identificándose 10 requisitos no funcionales y 23 requisitos funcionales agrupados en ocho casos de uso representados en el diagrama de casos de uso del sistema. Se modeló el diagrama de paquetes y el diagrama de clases del diseño con el fin de representar la distribución física y lógica de los principales objetos y clases que intervienen en el desarrollo de la aplicación. Además, se argumentó el uso de los patrones de casos de uso, arquitectónicos y de diseño que garantizan las buenas prácticas de diseño e implementación de la solución. En el modelo de datos se definieron las entidades usuario, proyecto, página y plantilla que almacenarán los atributos de los usuarios, los proyectos de libro web, las páginas con que cuentan los mismos y la información de las plantillas de libro web respectivamente. Finalmente se presentó el modelo de despliegue, que permite representar la ubicación física de los diferentes componentes que interactúan en el SGCLW.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DE LA SOLUCIÓN

Introducción

En el presente capítulo se presenta el modelo de implementación elaborado a partir de los resultados obtenidos en la fase de análisis y diseño llevada a cabo en el capítulo anterior. Se establecen el modelo de implementación y las pruebas de las funcionalidades del sistema. Las pruebas constituyen un elemento clave para garantizar la calidad del sistema y representan además una revisión final de todo el proceso de desarrollo.

3.1 Modelo de implementación

El modelo de implementación comprende el conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, clases, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe entre los elementos del modelo de diseño, y los subsistemas y componentes físicos (17).

3.1.1 Diagrama de componentes

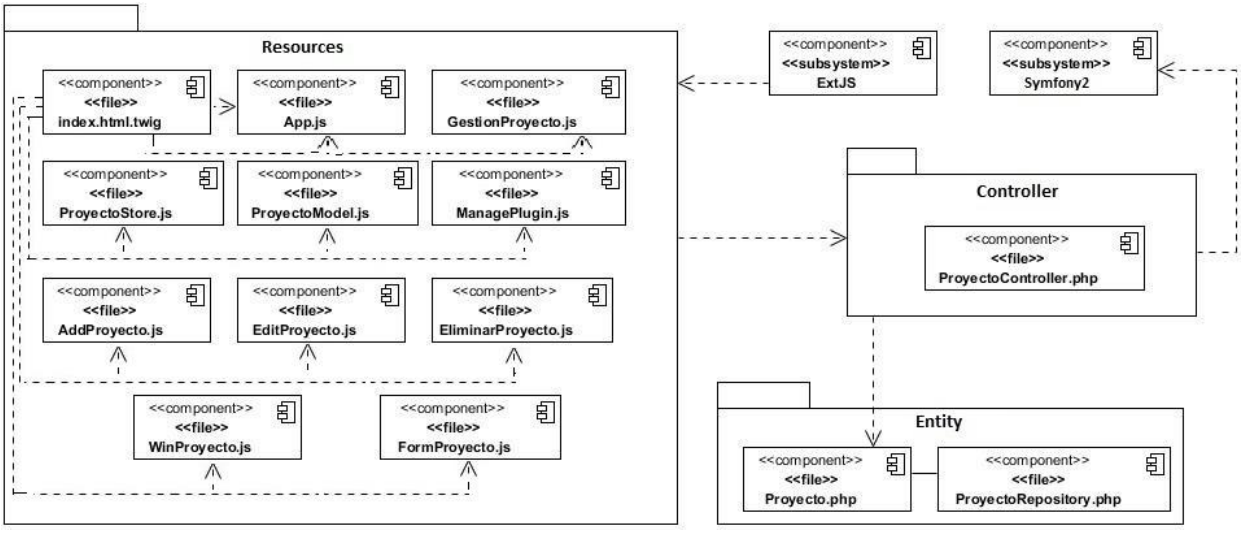


Figura 15 Diagrama de Componentes perteneciente al CU Gestionar PLW

El diagrama de componentes tiene como propósito modelar y representar los componentes de un software y sus dependencias. Para ello separa los diferentes archivos del código fuente en subsistemas y componentes. Cada subsistema representa un paquete o directorio del sistema, y cada componente representa un fichero o librería perteneciente al sistema. En la Figura 15 se muestra el diagrama de

componentes perteneciente al CU Gestionar PLW considerado arquitectónicamente significativo del sistema.

Descripción de los paquetes básicos de implementación:

- **Paquete *Resources*:** agrupa los componentes que permiten la interacción directa del usuario final con el sistema. Contiene las clases que componen la interfaz visual de la aplicación. Este paquete se corresponde con la capa de la vista del patrón arquitectónico MVC como se explica en el Patrón arquitectónico MVC
- **Paquete *Controller*:** contiene los componentes encargados de implementar las clases que manipulan las solicitudes del usuario y se apoya en el subsistema modelo para dar respuestas a las peticiones del mismo. Este paquete se corresponde con la capa controladora del patrón arquitectónico MVC.
- **Paquete *Entity*:** agrupa los componentes que implementan las clases encargadas del acceso a datos y la manipulación de los mismos. Según las capas que propone el patrón arquitectónico MVC este paquete se corresponde con la capa del modelo.

3.2 Código fuente

El código fuente de un sistema informático no es más que el lenguaje en que se representan las instrucciones que el ordenador debe ejecutar para llevar a cabo las funcionalidades del software. La presentación y estructura del código fuente varía según el lenguaje de programación empleado. Para almacenarlo de una manera organizada y comprensible, los desarrolladores hacen empleo de estándares de codificación que facilitan la lectura del código por parte de otros desarrolladores. La siguiente sección está destinada a explicar los estándares de codificación empleados durante el desarrollo del Sistema de Gestión de Contenidos para la Generación de Libros Web.

3.2.1 Estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a la estructura y apariencia física del código. Las normas de codificación definidas por la línea de arquitectura propuesta por el departamento Desarrollo de Aplicaciones están enfocadas a lograr una mejor comprensión y mantenimiento del código que responda a la complejidad dada por la cantidad de componentes y el alto nivel de integración que puede existir en el desarrollo del software. A continuación se enumeran algunas de las normas de codificación empleadas:

- Las variables locales serán inicializadas en el momento de su declaración o en caso contrario cuando su valor inicial dependa de una llamada a otro método en un determinado momento.
- Se utilizarán llaves de apertura y cierre en todos los casos de condiciones y ciclos para aumentar la legibilidad del código fuente.
- Se hará empleo de la nomenclatura CamelCase[18]. Las variables y métodos estarán escritos con el estándar lowerCamelCase, mientras que la declaración de clases hará empleo del estándar UpperCamelCase.
- Cada función debe tener un comentario, antes de su declaración, explicando qué hace. Tanto el nombre como el comentario que acompañe a la función deben bastar para ello.
- Todo el código implementado tendrá una indentación de cuatro espacios, facilitando la legibilidad y la comprensión del código.
- El límite máximo de longitud de una línea de código debe ser de 80 caracteres.

3.2.2 Ejemplos de código fuente

A continuación se presenta en la Figura 16 un fragmento de código donde se muestra el empleo de los estándares de codificación descritos en la sección anterior. El fragmento de código pertenece a la clase controladora ProyectoController.php y en él se observa el método listarAction que obtiene la lista de proyectos de libros web existentes en el sistema.

[18] CamelCase es un estilo de escritura que se aplica a frases o palabras compuestas. Posee dos variantes: UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula y lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula.

```
// (...)
class ProyectoController extends Controller {
 /*
    Método encargado de listar los proyectos de libro web
    Recive:     usuario propietario de los proyectos a listar
    Retorna:    listado de proyectos del usuario solicitado
*/
    /**
     *
     * @remote
     * @param array $params
     * @return array
     */
    public function listarAction($params) {
        $em = $this->getDoctrine()->getManager();
        $filter = $params['filter'][0]['value'];
        $entity = $em->getRepository('LibroswebBundle:Proyecto')->lsitarByUsuario($filter);
        $totalItems = count($entity);
        return array(
            'success' => true,
            'totalItems' => $totalItems,
            'items' => $entity
        );
    }

// (...)
```

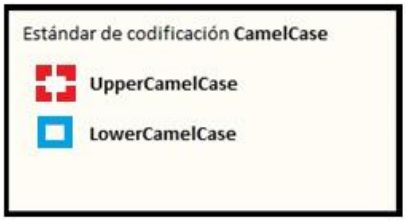


Figura 16 Ejemplo de empleo del estándar de codificación CamelCase

3.3 Estrategia de prueba del software

El flujo de trabajo de prueba consiste en la ejecución de un programa informático con la intención de descubrir errores. Las pruebas de software constituyen un elemento crítico para la garantía de calidad de un software. Estas permiten validar que el sistema implementado cumple con los requisitos definidos durante la fase de análisis y diseño (6). Las pruebas son aplicadas en diferentes momentos del ciclo de vida del software con el fin de comprobar hasta qué punto el programa desempeña un buen funcionamiento. Se considera que una prueba tiene éxito si detecta un error no identificado hasta el momento.

"*Los procesos de prueba para la ingeniería web comienzan con pruebas que ejercitan el contenido y la funcionalidad de la interfaz que es inmediatamente visible para los usuarios finales (...). Finalmente el foco cambia a las pruebas que ejercitan las capacidades tecnológicas que no siempre son aparentes a los usuarios*" (6). La Figura 17 ilustra el proceso de prueba de una aplicación web sobre la pirámide de diseño del sistema, según propone Roger S. Pressman. A medida que se realiza el flujo de pruebas de izquierda a derecha y de arriba hacia abajo, los elementos de la aplicación web visibles para el usuario se prueban primero, seguidos por los elementos de diseño de infraestructura del software.

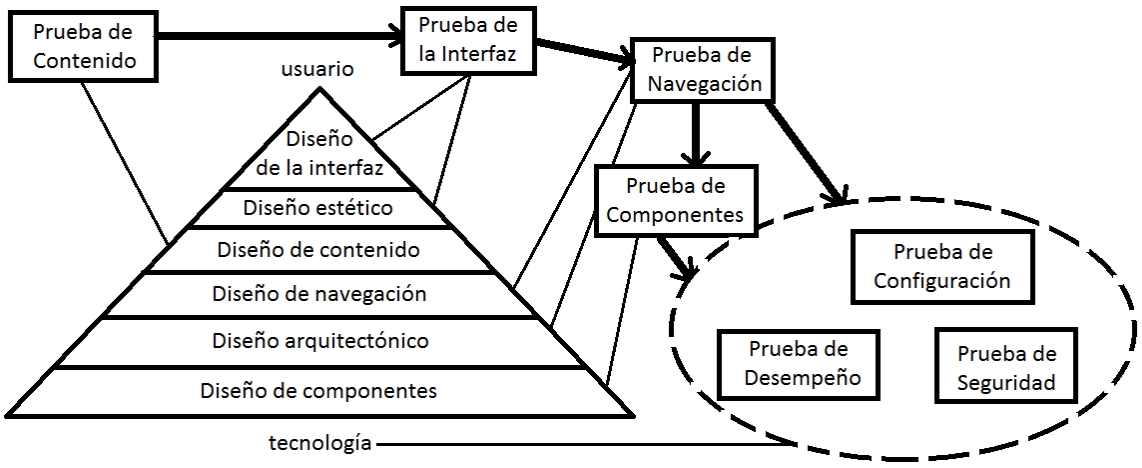


Figura 17 Proceso de prueba de una aplicación web

3.4 Pruebas de software

Las pruebas realizadas para validar el correcto funcionamiento del SGCLW emplean el **método de prueba**: Caja Negra, haciendo uso del tipo de prueba: Prueba Funcional y la técnica: Partición de Equivalencia. Este método conocido también como Prueba de Comportamiento, se ejecuta sobre cada caso de uso; definiendo casos de prueba que especifican un conjunto de entradas y respuestas esperadas del sistema ante cada una de ellas. En las siguientes secciones se explican cada una de las pruebas realizadas y se exponen los resultados asociados a cada una de ellas.

3.4.1 Prueba de contenido

Las pruebas de contenido (y las revisiones) intentan descubrir errores en el contenido de la aplicación. Estas pruebas permiten detectar errores tipográficos, equívocos gramaticales, errores en la consistencia del contenido, inexactitudes en las representaciones gráficas y fallas en las referencias cruzadas (6). De esta manera se garantiza que el contenido que presenta la aplicación al usuario es comprensible y no contiene errores.

3.4.2 Prueba de la interfaz

La prueba de la interfaz se ejercita sobre los mecanismos de interacción y valida los aspectos estáticos de la interfaz de usuario. El objetivo es descubrir los errores resultantes de mecanismos con una pobre implementación de interacción o ambigüedades introducidas en la interfaz de manera involuntaria.

3.4.3 Prueba de navegación

"*La prueba de navegación aplica casos de uso, derivados como parte de la actividad de análisis, en el diseño de casos de prueba que ejerciten cada escenario de uso contra el diseño de navegación*" (6). Los

mecanismos de navegación implementados dentro de la interfaz gráfica de la aplicación se prueban con el fin de garantizar que los errores que impiden completar el caso de uso se identifiquen y corrijan.

3.4.4 Prueba de componentes

La prueba de componentes ejercita el contenido y las unidades funcionales dentro de la aplicación web. Cada "unidad" corresponde a una interfaz de la aplicación la cual encierra contenidos, vínculos de navegación y elementos de procesamiento (6). A medida que se construye la arquitectura de la aplicación, las pruebas de navegación y de componentes se utilizan como pruebas de integración; permitiendo descubrir errores en las colaboraciones entre los diferentes componentes de la aplicación.

3.4.5 Prueba de configuración

"*Las pruebas de configuración intentan descubrir errores que son específicos respecto de un cliente o ambiente de servidor particulares*" (6). A través de la creación de una matriz de referencia cruzada que define los probables sistemas operativos y navegadores web, las pruebas se encaminan a descubrir errores asociados a cada posible configuración. Esta prueba persigue como principal objetivo determinar la configuración óptima del sistema.

3.4.6 Prueba de seguridad

La prueba de seguridad incorpora un conjunto de pruebas diseñadas para explotar las vulnerabilidades en una aplicación web y su ambiente (6). El objetivo principal de esta prueba es demostrar la existencia de una brecha de seguridad en la aplicación. Esta prueba permite reconocer si los roles definidos en el sistema y sus respectivos permisos funcionan de manera adecuada.

3.4.7 Prueba de desempeño

La prueba de desempeño abarca una serie de pruebas que permiten valorar cómo responde el sistema ante situaciones extremas. Estas pruebas permiten valorar los siguientes aspectos:

- ¿Cómo afecta el aumento de tráfico de usuarios el tiempo de respuesta?
- ¿Qué componentes del sistema son responsables de la degradación del desempeño y qué características de uso provocan que ocurra esta degradación?
- ¿Cómo la degradación del desempeño impacta los objetivos y requisitos de la aplicación? (6).

Para evaluar el desempeño del SGCLW se hará empleo de las pruebas de carga y estrés; que permiten verificar la aceptabilidad del desempeño del sistema ante condiciones extremas como grandes volúmenes de usuarios conectados o una carga excesiva de trabajo.

3.5 Diseño de casos de prueba

Los casos de pruebas se realizan con el objetivo de comprobar que una funcionalidad ha sido implementada satisfaciendo las necesidades del cliente. Asociado a cada caso de uso se confecciona un caso de prueba que recoja la especificación de requisitos de ese caso de uso, dividido en secciones y escenarios. Al mismo tiempo, se detallan las funcionalidades descritas en él y se describe cada variable de acuerdo al mismo. A continuación se presenta el caso de prueba perteneciente al CU Gestionar PLW considerado caso de uso arquitectónicamente significativo del sistema.

Tabla 13 Secciones del caso de prueba por método Caja Negra correspondiente al CU Gestionar PLW

Sección	Escenario	Descripción	Flujo Central
SC1: Gestionar PLW	**EC1** Gestionar PLW	El sistema muestra un listado con los proyectos de libros web pertenecientes al usuario autenticado.	1. El Usuario selecciona la opción **"Mis proyectos"** ubicada en el escritorio de la aplicación**.** 2. El sistema muestra la ventana **"Mis Proyectos"** con el listado de libros web pertenecientes al usuario autenticado.
SC2: Adicionar PLW	**EC2.1** Adicionar PLW correctamente.	El usuario inserta los datos del nuevo PLW y el sistema lo registra satisfactoriamente.	1. El usuario selecciona la opción **"Adicionar"** ubicada en la barra de herramientas de la ventana **"Mis Proyectos".** 2. El sistema muestra el formulario **"Propiedades del Proyecto"** con los campos del nuevo PLW. 3. El Usuario llena todos los datos del nuevo PLW y selecciona la opción **Aceptar.** 4. El sistema adiciona el nuevo proyecto y muestra el listado actualizado de PLWs.
	EC2.2: Adicionar PLW (dejando campos vacíos).	El sistema colorea de rojo los campos que estén vacíos y sean obligatorios.	1. El usuario selecciona la opción **"Adicionar"** ubicada en la barra de herramientas de la ventana **"Mis Proyectos"**. 2. El sistema muestra el formulario **"Propiedades del Proyecto"** con los campos del nuevo PLW. 3. El Usuario no llena todos los campos y selecciona la opción **Aceptar.** 4. El sistema colorea los campos vacíos.

SC3 Modificar PLW	**EC3.1** Modificar PLW correctamente.	El sistema modifica las propiedades de un PLW si los elementos insertados son válidos.	1. El usuario selecciona un PLW y elige la opción **"Editar"** ubicada en la barra de herramientas de la ventana **"Mis Proyectos"**. 2. El sistema muestra el formulario **"Propiedades del Proyecto"** con campos que contienen los valores del PLW seleccionado. 3. El Usuario modifica los datos del PLW y selecciona la opción **Aceptar.** 4. El sistema modifica el proyecto y muestra el listado actualizado de PLWs.
	EC3.2 Modificar Libro Web (dejando campos vacíos).	El sistema colorea de rojo los campos que estén vacíos y sean obligatorios.	1. El usuario selecciona un PLW y elige la opción **"Editar"** ubicada en la barra de herramientas de la ventana **"Mis Proyectos"**. 2. El sistema muestra el formulario **"Propiedades del Proyecto"** con campos que contienen los valores del PLW seleccionado. 3. El Usuario modifica los datos del PLW dejando al menos un campo vacío y selecciona la opción **Aceptar.** 4. El sistema colorea los campos vacíos.
SC4 Eliminar PLW	**EC4.1** Eliminar PLW satisfactoriamente.	El sistema elimina el PLW seleccionado por el usuario.	1. El usuario selecciona un proyecto de la lista ubicada en la ventana **"Mis Proyectos"** y seguidamente selecciona la opción **"Eliminar"** ubicada en la barra de herramientas. 2. El sistema muestra un mensaje de confirmación pidiendo afirmación para eliminar el PLW. 3. El usuario selecciona la opción **"Sí"**. 4. El sistema elimina el proyecto y muestra el listado actualizado de PLWs.
	EC4.2 Eliminar PLW (cancelar operación)	El sistema cancela la eliminación de un PLW en caso de que el usuario desista de	1. El usuario selecciona un proyecto de la lista ubicada en la ventana **"Mis Proyectos"** y seguidamente ejecuta la opción **"Eliminar"** ubicada en la barra de herramientas de la ventana **"Mis Proyectos"**.

		eliminar un PLW.	2. El sistema muestra un mensaje de confirmación pidiendo afirmación para eliminar el libro web. 3. El usuario selecciona la opción **"No"**. 4. El sistema no elimina el proyecto y el listado actualizado de PLWs.
SC5 Transferir PLW	**EC5.1** Transferir PLW satisfactoriamente	El sistema transfiere un PLW de un usuario a otro cambiando el propietario del mismo.	1. El Usuario selecciona un proyecto y ejecuta la opción **"Transferir"** ubicada en la barra de herramientas de la interfaz **"Mis proyectos"**. 2. El sistema muestra la interfaz **"Transferir Proyecto"** que contiene un campo de selección que permite elegir el correo electrónico de los usuarios del sistema. 3. El usuario selecciona el correo electrónico del usuario al que desea transferir el PLW y selecciona la opción **"Aceptar"**. 4. El sistema transfiere el PLW al usuario seleccionado y muestra el listado actualizado de PLWs.
	EC5.1 Transferir PLW (cancelar operación)	El sistema cancela la transferencia de un PLW si el usuario selecciona la opción Cancelar.	1. El Usuario selecciona la opción **"Transferir"** ubicada en el la barra de herramientas de la interfaz **"Mis proyectos"**. 2. El sistema muestra la interfaz **"Transferir Proyecto"** que contiene un campo de selección que permite elegir el correo electrónico de los usuarios del sistema. 3. El usuario selecciona la opción **"Cancelar"**. 4. El sistema transfiere el PLW al usuario seleccionado y muestra el listado actualizado de PLWs.
SC6 Visualizar PLW	**EC6.1** Visualizar Libro Web	El sistema muestra los detalles de un proyecto de libro web.	1. El usuario selecciona un libro web de la lista de proyectos ubicada en la ventana **"Mis Proyectos"**. 2. El Usuario selecciona la opción denotada con el símbolo "+" en el libro web seleccionado. 3. El elemento de la lista despliega una vista detallada del proyecto de libro web seleccionado.

Tabla 14 Descripción de las variables presentes en caso de prueba analizado.

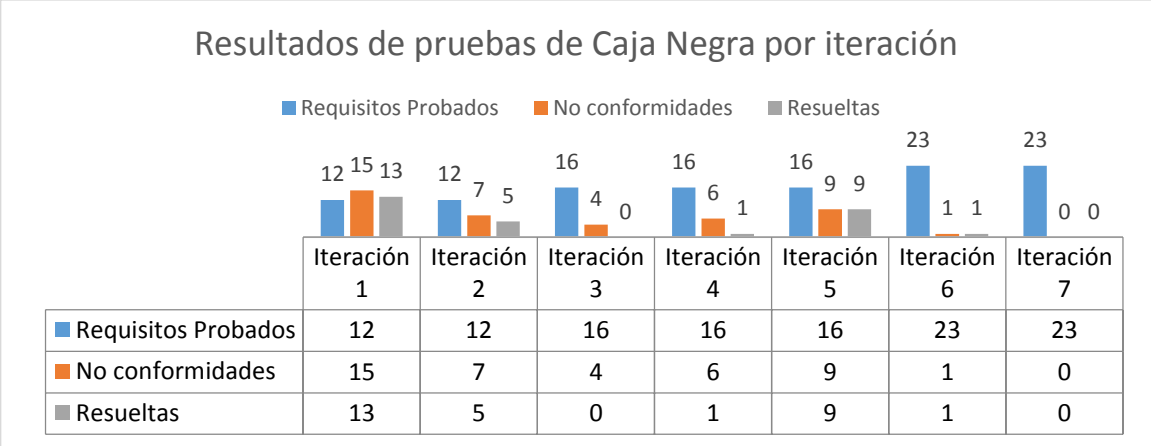
No	Nombre Campo	Clasificación	Valor Nulo	Descripción
1	Título	Campo de texto	No	Cadena de texto (admite hasta 255 caracteres).
2	Subtítulo	Campo de texto	No	Cadena de texto (admite hasta 255 caracteres).
3	Autor	Campo de texto	No	Cadena de texto (admite hasta 255 caracteres).
4.	Descripción	Campo de texto	No	Cadena de texto (admite hasta 500 caracteres).
5.	Aceptar	Botón	No	Permite que el usuario acepte una operación.
6	Cancelar	Botón	No	Permite que el usuario Cancele una operación.
7	Correo	Casilla de selección	No	Permite seleccionar el correo electrónico de un usuario.

3.6 Resultados de las pruebas de software

En el presente epígrafe se describen los resultados generales de las pruebas aplicadas al SGCLW definidas en el epígrafe Pruebas de software. Se especifican los diferentes ambientes de ejecución que se tomaron en cuenta para llevar a cabo las pruebas de configuración.

Pruebas Funcionales por método de Caja Negra

Una vez concluidas las pruebas funcionales haciendo empleo de la técnica partición de equivalencia y guiadas por los casos de prueba, se verificó el correcto funcionamiento del sistema al satisfacer todos los requisitos funcionales definidos. La realización de las pruebas permitió identificar los errores no detectados hasta entonces. En la Figura 18 se muestran los resultados obtenidos a lo largo de siete iteraciones, para cada iteración se especifican los requisitos evaluados durante las pruebas, así como las no conformidades existentes y resueltas al final de cada iteración.



	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Iteración 5	Iteración 6	Iteración 7
Requisitos Probados	12	12	16	16	16	23	23
No conformidades	15	7	4	6	9	1	0
Resueltas	13	5	0	1	9	1	0

Figura 18 Resultados de las pruebas de Caja Negra

Las pruebas de Caja Negra permitieron llevar a cabo las pruebas de contenido, de interfaz, de navegación de componentes, de configuración y de seguridad reflejadas en la Figura 17 Proceso de prueba de una aplicación web.

Pruebas de configuración

La Tabla 15 es una matriz de referencia cruzada donde se establecen nomenclaturas para las posibles configuraciones que describen los escenarios probables donde se ejecutará la aplicación desarrollada. Estas configuraciones dependen del sistema operativo y el navegador web del ordenador desde donde se empleará la aplicación. Para probar el correcto desempeño del sistema en diferentes entornos, cada una

de las iteraciones de las pruebas de Caja Negra fue llevada a cabo empleando una de las configuraciones establecidas. Esta matriz permite referenciar dichas configuraciones en caso de detectar una no conformidad relacionada a las mismas (Ejemplo: Si existiera una no conformidad asociada a la ejecución del SGCLW desde el sistema operativo Ubuntu empleando el navegador web Firefox, esta no conformidad estaría asociada a la configuración CF1).

Tabla 15 Nomenclatura de las configuraciones Sistema Operativo - Navegador Web a probar

		Navegador Web		
		Firefox	**Chrome**	**Internet Explorer**
Sistema Operativo	**Ubuntu**	CF1	CF2	-
	Windows	CF3	CF4	CF5
	Nova	CF6	CF7	-

Una vez realizadas las pruebas funcionales, teniendo en cuenta los posibles escenarios donde puede ser ejecutada la aplicación, se comprobó que la aplicación funciona correctamente para cada una de las configuraciones establecidas.

Pruebas de seguridad

Las pruebas de seguridad fueron realizadas a través de la interacción entre la aplicación y varios usuarios con diferentes roles. Empleando la técnica de Caja Negra se pudieron analizar las respuestas del sistema ante un conjunto de entradas definidas en los casos de prueba. Estas pruebas permitieron validar que cada usuario solo tiene acceso a sus proyectos de libro web asegurando de esta manera la confidencialidad de la información. También pudo validarse durante el desarrollo de estas pruebas que los diferentes roles solo tienen acceso a las funcionalidades requeridas. Las pruebas de seguridad ejecutadas a lo largo de siete iteraciones no arrojaron no conformidades.

Pruebas de rendimiento (Carga)

Según el autor Jakob Nielsen, en el libro "*Usability Engineering*" (21) existen tres límites importantes en el tiempo de respuesta de una aplicación web:

- 0,1 segundo: es el límite en el cual el usuario siente que está "manipulando" los objetos de la interfaz de usuario.
- 1 segundo: es el límite en el cual el usuario siente que está navegando libremente sin esperar demasiado una respuesta del servidor.

- 10 segundos: es el límite en el cual se pierde la atención del usuario.

Para llevar a cabo las pruebas de rendimiento se estableció como tiempo aceptable de respuesta de la aplicación: dos segundos, y se estimó que el sistema tendría como máximo 50 conexiones de usuarios trabajando simultáneamente. Una vez estimados los límites de conexiones y establecido el tiempo aceptable de respuesta, se procede a la realización de las pruebas de rendimiento haciendo empleo de la herramienta Apache Jmeter en su versión 2.12.

En la Tabla 16 se resume el análisis de los resultados obtenidos durante las pruebas de rendimiento haciendo empleo de la herramienta Jmeter. La aplicación respondió ante 50 conexiones recurrentes de usuario con un tiempo de respuesta máximo de 0.9 segundo. Una segunda prueba arrojó un tiempo de respuesta máximo de 1.2 segundos para 100 conexiones de usuarios realizando solicitudes al sistema simultáneamente. Concluyéndose que la aplicación supera los límites de usuarios sin rebasar los tiempos de respuesta establecidos.

Tabla 16 Resultados de las pruebas de rendimiento con la herramienta Jmeter

Muestras	Solicitudes	Tiempo de respuesta
50	50	934 ms
100	100	1200 ms

Conclusiones parciales

Las tareas de la investigación asociadas a la fase de implementación y prueba permitieron la creación del modelo de implementación, que permitió representar la composición física de los componentes y subsistemas del SGCLW. Se identificaron los estándares y normas de codificación, que garantizan la legibilidad y facilitan la comprensión del código fuente de la aplicación. Durante la fase de pruebas de software se definió la aplicación de pruebas de Caja Negra guiadas por casos de prueba, realizando pruebas funcionales haciendo uso de la técnica partición de equivalencia. Las pruebas de configuración validaron que la aplicación desarrollada es compatible con los sistemas operativos y navegadores web empleados en la universidad. Durante las pruebas de seguridad se validó que la información gestionada en la aplicación sólo es accesible por su respectivo propietario y los mecanismos de permisos aseguran que las funcionalidades del sistema sólo son ejecutadas por los roles autorizados. Las pruebas de Caja Negra se ejecutaron en siete iteraciones durante las cuales se probó el cumplimiento de los requisitos del sistema. Fueron encontradas un total de 42 no conformidades, todas resueltas al término de la última iteración.

CONCLUSIONES GENERALES

Una vez culminado el proceso de desarrollo del Sistema de Gestión de Contenidos para la Generación de Libros Web se puede afirmar que se dio cumplimiento a los objetivos propuestos al inicio de la investigación. Para ello:

- Se determinó el uso del editor de texto HTML CKEditor, el cual permite la generación y edición de texto en formato HTML de manera transparente para el usuario. Además de brindar al usuario a través de la interfaz visual del mismo una experiencia similar al trabajo con herramientas como Microsoft Word o Libre Office.
- Teniendo en cuenta las exigencias y necesidades del centro DATEC relacionadas a la creación de manuales de ayuda, fueron identificados 10 requisitos no funcionales y 23 requisitos funcionales agrupados en ocho casos de uso. La aplicación de patrones de diseño y el empleo del patrón arquitectónico MVC, facilitaron el desarrollo del sistema aportando calidad al producto final.
- Se implementaron los requisitos funcionales identificados, obteniéndose una aplicación capaz de gestionar contenidos en formato HTML para la posterior generación de libros web. El sistema además permite importar un libro web que ya no esté disponible en la aplicación, para llevar a cabo una nueva edición partiendo del contenido existente en él. A través del empleo de estándares de codificación se garantizó la legibilidad y organización del código fuente de la aplicación. Como resultado se obtuvo una aplicación web escalable, que puede ser expandida a través de la creación de módulos sin afectar la calidad de los servicios.
- Se validaron las funcionalidades implementadas a través de la interfaz del sistema, mediante el diseño de pruebas de Caja Negra guiado por casos de prueba y aplicando la técnica partición de equivalencia. Las pruebas permitieron detectar errores en el comportamiento y en el contenido de la aplicación. Los principales errores detectados durante las siete iteraciones estuvieron relacionados a los mecanismos de navegación y uso incorrecto de los componentes del marco de trabajo ExtJS. Finalmente fueron resueltas todas las no conformidades al término de la última iteración de pruebas, alcanzado la correcta implementación de cada requisito del sistema.

RECOMENDACIONES

Se recomienda incrementar las funcionalidades del Sistema de Gestión de Contenidos para la Generación de Libros Web, de manera tal que permita la disponibilidad en línea de libros web en la Universidad de las Ciencias Informáticas. Para la implementación de este sistema se recomiendan las siguientes tareas:

- Implementar un módulo para el Sistema de Gestión de Contenidos para la Generación de Libros Web que permita establecer estados sobre los PLW. Permitiendo al administrador revisar y aprobar la publicación de libros a petición de los usuarios que los han creado.
- Implementar un mecanismo que permita etiquetar los libros web existentes en el sistema con el fin de establecer categorías sobre los mismos según el tópico que tratan.
- Implementar un módulo con interfaz visual que permita realizar búsquedas sobre los libros web publicados en el sistema a partir de palabras claves, a través de las etiquetas asignadas a los libros o combinando estos dos métodos. Se sugiere que esta interfaz no requiera autenticación y presente una apariencia similar a los buscadores web.
- Por último se propone fomentar el uso del Sistema de Gestión de Contenidos para la Generación de Libros Web no solo para la creación y publicación de manuales de ayuda; sino también para la publicación de tutoriales y documentos útiles en general para la comunidad universitaria.

REFERENCIAS BIBLIOGRÁFICAS

1. **Salazar, Pedro Román.** Observatorio Tecnológico. [En línea] 19 de Enero de 2011. [Citado el: 8 de Octubre de 2014.] http://recursostic.educacion.es/observatorio/web/es/equipamiento-tecnologico/hardware/954-libros-electronicos-ebooks-.

2. **DeConceptos.com.** DeConceptos. [En línea] 4 de Noviembre de 2012. [Citado el: 8 de Octubre de 2014.] http://deconceptos.com/tecnologia/manual-de-usuario.

3. **Robohelp.** Adobe Systems. [En línea] [Citado el: 20 de Octubre de 2014.] http://www.adobe.com/es/products/robohelp.

4. **Dr.Explain.** Dr.Explain. [En línea] Indigo Byte Systems. [Citado el: 3 de Diciembre de 2014.] http://www.drexplain.es/.

5. **Sigil.** Sigil. [En línea] [Citado el: 3 de Diciembre de 2014.] https://code.google.com/p/sigil/.

6. **Pressman, Roger S.** *Ingenieria de Software. Un enfoque práctico. Sexta edición.* La Habana : Felix Varela, 2005.

7. **Eclipse.org.** OpenUP. [En línea] Eclipse, 2012. [Citado el: 3 de Octubre de 2014.] http://epf.eclipse.org/wikis/openup/.

8. **Object Management Group.** Object Management Group. [En línea] 2009 [Citado el: 3 de Octubre de 2014.] http://www.uml.org/.

9. **Scribd.** Scribd. [En línea] 2011. [Citado el: 20 de Octubre de 2014.] http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE.

10. **RENa.** [En línea] 2008. [Citado el: 3 de Diciembre de 2014.] http://www.rena.edu.ve/cuartaEtapa/Informatica/Tema13.html.

11. Programacinweb.net. Programacinweb.net. [En línea] 2009. [Citado el: 12 de Octubre de 2014.] http://www.programacionweb.net/articulos/articulo/?num=686.

12. Librosweb.es. LIBROSWEB. [En línea] 2013. [Citado el: 3 de Octubre de 2014.] http://librosweb.es/javascript/.

13. NetBeans.org. NetBeans. [En línea] 2014 [Citado el: 3 de Octubre de 2014.] https://netbeans.org/.

14. Estudioteca. Estudioteca. [En línea] 2012. [Citado el: 20 de Octubre de 2014.] http://www.estudioteca.net/universidad/telecomunicaciones/gestor-base-datos/..

15. Symfony. Symfony. [En línea] 2013. [Citado el: 20 de Octubre de 2014.] http://symfony.es/que-es-symfony..

16. CKEditor. CKEditor. [En línea] 2013. [Citado el: 20 de Octubre de 2014.] http://www.ckeditor.org.

17. Larman, Craig. *UML y Patrones.* México : Prentice Hall, 1999. ISBN: 970-17-0261-1.

18. Sommerville, Ian. *Ingeniería de Software. Séptima Edición.* 2005. 84-7829-074-5.

19. Buschmann, Frank, y otros. *Pattern Oriented Software Architecture.* s.l. : Wiley, 1999. 0 471 95889 7.

20. Patrones de Diseño. [En línea] 2009. [Citado el: 20 de Octubre de 2014.] http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf..

21. Nielsen, Jakob. *Usability Engineering.* San Francisco : Morgan Kaufmann. ISBN 0125184069.

22. Ptrones. [En línea] 2007. [Citado el: 25 de Octubre de 2014.] http://www.info-ab.uclm.es/asignaturas/42579/pdf/04-pitulo4a.pdf.

23. PostgreSQL. PostgreSQL. [En línea] 2014. [Citado el: 3 de Octubre de 2014.] http://www.postgresql.org/about/.

24. Sencha. Sencha. [En línea] Sencha, 2014. [Citado el: 3 de Octubre de 2014.] http://www.sencha.com/products/extjs/.

25. Sencha. Sencha Docs. [En línea] Sencha, 2014. [Citado el: 3 de Octubre de 2014.] docs.sencha.com/extjs/4.2.0/.

26. Agile Modeling. [En línea] Ambysoft Inc, 2014. [Citado el: 6 de Marzo de 2015.] http://www.agilemodeling.com/artifacts/useCaseDiagram.htm.

BIBLIOGRAFÍA

1. **Salazar, Pedro Román.** Observatorio Tecnológico. [En línea] 19 de Enero de 2011. [Citado el: 8 de Octubre de 2014.] http://recursostic.educacion.es/observatorio/web/es/equipamiento-tecnologico/hardware/954-libros-electronicos-ebooks-.

2. **DeConceptos.com.** DeConceptos. [En línea] 4 de Noviembre de 2012. [Citado el: 8 de Octubre de 2014.] http://deconceptos.com/tecnologia/manual-de-usuario.

3. **Robohelp.** Adobe Systems. [En línea] [Citado el: 20 de Octubre de 2014.] http://www.adobe.com/es/products/robohelp.

4. **Dr.Explain.** Dr.Explain. [En línea] Indigo Byte Systems. [Citado el: 3 de Diciembre de 2014.] http://www.drexplain.es/.

5. **Sigil.** Sigil. [En línea] [Citado el: 3 de Diciembre de 2014.] https://code.google.com/p/sigil/.

6. **Pressman, Roger S.** *Ingenieria de Software. Un enfoque práctico. Sexta edición.* La Habana : Felix Varela, 2005.

7. **Eclipse.org.** OpenUP. [En línea] Eclipse, 2012. [Citado el: 3 de Octubre de 2014.] http://epf.eclipse.org/wikis/openup/.

8. **Object Management Group.** Object Management Group. [En línea] 2009 [Citado el: 3 de Octubre de 2014.] http://www.uml.org/.

9. **Scribd.** Scribd. [En línea] 2011. [Citado el: 20 de Octubre de 2014.] http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE.

10. **RENa.** [En línea] 2008. [Citado el: 3 de Diciembre de 2014.] http://www.rena.edu.ve/cuartaEtapa/Informatica/Tema13.html.

11. Programacinweb.net. Programacinweb.net. [En línea] 2009. [Citado el: 12 de Octubre de 2014.] http://www.programacionweb.net/articulos/articulo/?num=686.

12. Librosweb.es. LIBROSWEB. [En línea] 2013. [Citado el: 3 de Octubre de 2014.] http://librosweb.es/javascript/.

13. NetBeans.org. NetBeans. [En línea] 2014 [Citado el: 3 de Octubre de 2014.] https://netbeans.org/.

14. Estudioteca. Estudioteca. [En línea] 2012. [Citado el: 20 de Octubre de 2014.] http://www.estudioteca.net/universidad/telecomunicaciones/gestor-base-datos/..

15. Symfony. Symfony. [En línea] 2013. [Citado el: 20 de Octubre de 2014.] http://symfony.es/que-es-symfony..

16. CKEditor. CKEditor. [En línea] 2013. [Citado el: 20 de Octubre de 2014.] http://www.ckeditor.org.

17. Larman, Craig. *UML y Patrones.* México : Prentice Hall, 1999. ISBN: 970-17-0261-1.

18. Sommerville, Ian. *Ingeniería de Software. Séptima Edición.* 2005. 84-7829-074-5.

19. Buschmann, Frank, y otros. *Pattern Oriented Software Architecture.* s.l. : Wiley, 1999. 0 471 95889 7.

20. Patrones de Diseño. [En línea] 2009. [Citado el: 20 de Octubre de 2014.] http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf..

21. Nielsen, Jakob. *Usability Engineering.* San Francisco : Morgan Kaufmann. ISBN 0125184069.

22. Ptrones. [En línea] 2007. [Citado el: 25 de Octubre de 2014.] http://www.info-ab.uclm.es/asignaturas/42579/pdf/04-pitulo4a.pdf.

23. PostgreSQL. PostgreSQL. [En línea] 2014. [Citado el: 3 de Octubre de 2014.] http://www.postgresql.org/about/.

24. Sencha. Sencha. [En línea] Sencha, 2014. [Citado el: 3 de Octubre de 2014.] http://www.sencha.com/products/extjs/.

25. Sencha. Sencha Docs. [En línea] Sencha, 2014. [Citado el: 3 de Octubre de 2014.] docs.sencha.com/extjs/4.2.0/.

26. Agile Modeling. [En línea] Ambysoft Inc, 2014. [Citado el: 6 de Marzo de 2015.] http://www.agilemodeling.com/artifacts/useCaseDiagram.htm.