

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1



TÍTULO: Plataforma colaborativa para la publicación de documentos digitales.

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

AUTORES: Adriam Balceiro Rodríguez

Alejandro Almarales Claro

TUTORES: Ing. Eduardo Manuel Macías Sotolongo

Ing. Paúl Rodríguez Leyva

Ciudad de La Habana, 17 de Junio 2013

“Año 54 de la Revolución”

DEDICATORIA

ALEJANDRO:

“A toda mi familia, en especial a mis padres Efrén Almarales Isaac y Noemí Claro Camiñas y a mi primo Alberto Martínez Claro.”

ADRIAM:

“A mi mamá y mi hermano, Maida Caridad Rodríguez Martínez y Ariam Balceiro Rodríguez.”

“A mi tía y mi tío, Magali Rodríguez Martínez y Ismael de Jesús Balceiro Sánchez.”

“A mis primos Isleam Balceiro Rodríguez y Leisam Balceiro Rodríguez.”

“A mi abuela y Adelina.”

“A mi grandísima familia.”

“A mis buenos y malos amigos.”

AGRADECIMIENTOS

ALEJANDRO:

A mis padres porque siempre han estado cuando más los he necesitado y por apoyarme siempre.

A mi primo Alberto que más que mi primo es mi hermano por soportarme siempre y complacerme en todo.

A mi familia por su apoyo y su cariño.

A la Revolución por darme la posibilidad de estar aquí hoy.

A mi maestra Natacha que creó en mí la vocación por la informática.

A los profesores que de una forma u otra han sabido hacer de mí lo que soy hoy, en especial a los que quedan para toda la vida: Liliam, Eugenio, Aleida, Jorge Luis y Quirós.

A mis amigos del IPVCE: Anamarys, Linnet, Lenier, Celia, Luis Daniel, Jose Carlos, Ernesto, Maidiel, Arley y Yagniel.

A mis compañeros del grupo y de la universidad con los que he compartido tantos momentos en estos cinco años.

A mis compañeros de apartamento: Angel, Alex, Osvaldo, Alfredo, Veitía, Luis Daniel y en especial a Nodeluis por aguantarme estos cinco años.

A mi compañero de tesis Adriam que sin su colaboración no hubiese sido posible la realización de este trabajo.

A mis tutores Paul y Adriam que son parte importante de este resultado.

A la gente del laboratorio 18 por incitarme siempre a trabajar, en especial a Samuel, Leo y Conde.

A Melisa por hacerme sonreír con sus travesuras y cambiarme la forma de ver todo.

AGRADECIMIENTOS

ADRIAM:

A nuestros tutores Ing. Eduardo Manuel Macías Sotolongo y Ing. Paúl Rodríguez Leyva, por orientarnos, ayudarnos y dedicar tanto tiempo y empeño en realizar con calidad el desarrollo de este trabajo.

A Alejandro Almarales mi compañero de tesis y amigo, por compartir su esfuerzo y conocimiento.

A la universidad y todos los profesores que han contribuido a mi formación y enseñanza.

A mis amigos y amigas de la universidad con los compartí momentos de estrés, ocio, alegría y experiencias inolvidables. A mis compañeros de grupo, de juegos y del lab 18.

A toda mi familia: tías y tíos, primos y primas, mi abuela, por su apoyo incondicional, paciencia y confianza en todos los momentos que pasamos juntos.

A mis pocos amigos del barrio.

A mi guía y hermano Ariam Balceiro Rodríguez y mi madre Maida Caridad Rodríguez Martínez, que nunca se han dado por vencidos en la tarea de educarme y encaminarme en la vida.

Al amor de mi vida y única sobrina Aliannys Balceiro García.

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 _____.

Firma del Autor

Adriam Balceiro Rodríguez

Firma del Autor

Alejandro Almarales Claro

Firma del Tutor

Ing. Paúl Rodríguez Leyva

RESUMEN

Materiales que son el resultado de conferencias científicas y seminarios, cotidianos en entornos académicos, no cuentan con un espacio donde puedan compartirse y enriquecerse con los criterios de los cibernautas. La inexistencia de estos espacios influye en que las obras literarias cubanas no tienen presencia y posicionamiento en Internet, lo que dificulta su difusión dentro y fuera del país. También es importante señalar la dificultad a la que se enfrentan las personas para compartir en la intranet documentos que no cuentan con la formalidad necesaria para ser publicados en los espacios existentes, pero que no dejan de constituir un aporte en determinadas ramas de la ciencia y/o la cultura. Por lo que el presente trabajo se ha trazado como objetivo general: Desarrollar una plataforma colaborativa mediante la informatización de los procesos de divulgación y recuperación de documentos digitales para mejorar su publicación y disponibilidad en la intranet cubana. Para el desarrollo del sistema se decidió emplear: la metodología de desarrollo OpenUp, el lenguaje de modelado UML, los lenguajes de codificación HTML, JavaScript, CSS y PHP, como marco de trabajo Symfony 2.0 y MongoDB como gestor de base de datos. El estudio del estado del arte permitió desarrollar una aplicación basada en las buenas prácticas empleadas por los sistemas estudiados y enfocada hacia la solución del problema identificado. Como resultado del trabajo realizado se logró el desarrollo de una plataforma colaborativa para la publicación de documentos digitales lo que mejorará la disponibilidad de este tipo de documentos en la intranet nacional.

Palabras claves: Plataforma colaborativa, documento, publicación, recuperación.

ÍNDICE

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: Fundamentación teórica	6
1.1. Descripción general del objeto de estudio	6
1.2. Sistemas similares.....	8
1.2.1. Google Books	8
1.2.2. ComparteLibros.....	9
1.2.3. LibrosGratis.....	9
1.2.4. Biblioteca Digital de la Biblioteca Nacional José Martí	10
1.2.5. Sunshine.....	10
1.2.6. Orión.....	11
1.3. Metodologías de desarrollo.....	13
1.3.1. AUP (Agile Unified Process).....	14
1.3.2. SCRUM	14
1.3.3. XP	15
1.3.4. OpenUP	16
1.4. Lenguaje de modelado.....	17
1.5. Lenguajes de codificación.....	18
1.5.1. Lenguaje de Marcado de Hipertexto HTML	18
1.5.2. Hojas de Estilo en Cascada	19
1.5.3. JavaScript	19
1.5.4. PHP.....	20
1.6. Marco de trabajo	20
1.7. Sistema Gestor de Base de Datos	21
1.7.1. PostgreSQL	22
1.7.2. MongoDB	22
1.8. Servidor web	23
1.9. Herramientas	24
1.9.1. Herramienta CASE de modelado	24

ÍNDICE

1.9.2. Netbeans 7.2.....	25
1.10. Conclusiones	25
Capítulo #2: Análisis y diseño de la plataforma colaborativa para la publicación de documentos digitales.....	27
2.1. Propuesta del sistema	27
2.2. Modelo de dominio.....	27
2.3. Relación de los requerimientos del sistema	29
2.3.1. Técnicas de captura de requisitos	29
2.3.2. Requerimientos funcionales	29
2.3.3. Requerimientos no funcionales	31
2.4. Modelo de casos de uso del sistema.....	31
2.4.1. Definición de los actores del sistema.....	31
2.4.2. Definición de los casos de uso del sistema	32
2.4.3. Patrones de casos de uso	33
2.4.4. Descripción de los casos de uso del sistema	35
2.5. Arquitectura del sistema	37
2.6. Diagrama de clases del diseño	38
2.6.1. Patrones de diseño empleados durante el desarrollo del sistema	41
2.7. Diagramas de interacción del diseño	44
2.8. Diseño de la base de datos.....	45
2.9. Diagrama de despliegue.....	47
2.10. Conclusiones	47
CAPÍTULO 3: Implementación y prueba de la plataforma colaborativa para la publicación de documentos digitales.....	48
3.1. Estándares de codificación	48
3.2. Diagrama de componentes.....	51
3.3. Interfaces principales de la aplicación	54
3.4. Pruebas de software.....	56
3.4.1. Pruebas de funcionalidad.....	57
3.4.2. Pruebas unitarias y de integración	61
3.4.3. Pruebas de carga y estrés	62
3.4.4. Pruebas de seguridad	62

ÍNDICE

3.5. Conclusiones	63
Conclusiones Generales	64
Recomendaciones.....	65
Referencias Bibliográficas	66
Bibliografía.....	69
Glosario de Términos	70
Anexos	71

ÍNDICE

ÍNDICE DE TABLAS

Tabla 1. Características presentes en los sistemas estudiados	13
Tabla 2. Justificación de los actores del sistema.	32
Tabla 3. Descripción CU Proponer documento.	36
Tabla 4. Descripción CU Registrar usuario.	37
Tabla 5. Descripción de las variables.	58
Tabla 6. Caso de prueba del CU Proponer Documento.	59
Tabla 7. Descripción de las variables.	59
Tabla 8. Caso de prueba del CU Registrar Usuario.	60

ÍNDICE

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Diagrama de clases del modelo de dominio.....	28
Ilustración 2. Diagrama de casos de uso.....	33
Ilustración 3. Aplicación del patrón reuso.....	34
Ilustración 4. Aplicación del patrón extensión.....	34
Ilustración 5. Diagrama de clases del diseño CU proponer documento.....	39
Ilustración 6. Diagrama de clases del diseño CU registrar usuario.....	40
Ilustración 7. Aplicación del patrón experto.....	42
Ilustración 8. Aplicación del patrón creador.....	43
Ilustración 9. Diagrama de secuencia CU proponer documento.....	44
Ilustración 10. Diagrama de secuencia CU registrar usuario.....	45
Ilustración 11. Modelo físico de datos para MongoDB.....	46
Ilustración 12. Diagrama de despliegue.....	47
Ilustración 13. Diagrama de componentes.....	52
Ilustración 14. Diagrama de componentes del bundle de administración.....	52
Ilustración 15. Diagrama de componentes del bundle de búsqueda.....	53
Ilustración 16. Diagrama de componentes del bundle de temas.....	53
Ilustración 17. Diagrama de componentes del bundle de comentarios.....	53
Ilustración 18. Diagrama de componentes del bundle de documentos.....	53
Ilustración 19. Diagrama de componentes del bundle de usuario.....	54
Ilustración 20. Interfaz para la búsqueda de documentos con filtros.....	54
Ilustración 21. Resultados de búsquedas paginado.....	55
Ilustración 22. Interfaz para la gestión de usuarios y permisos.....	55
Ilustración 23. Interfaz que muestra la información de un documento.....	56
Ilustración 24. No conformidades detectadas distribuidas por iteraciones.....	61
Ilustración 25. Tipos de no conformidades detectadas.....	61

INTRODUCCIÓN

Desde los inicios mismos de la civilización humana, el hombre aprendió a escribir para legar a las generaciones futuras sus conocimientos y experiencias. En los libros se encuentra una fuente de cultura vital para los pueblos y fuente de conocimiento esencial para el desarrollo científico técnico de la humanidad. El uso de documentos digitales no es reciente, la utilización de procesadores de texto data de hace muchos años, pero el uso de nuevas tecnologías de comunicación, especialmente Internet, junto con el incremento de los costos de producción de los medios impresos, ha generado un cambio en el concepto de publicación. Ahora es posible ahorrarse los costos de impresión, de encuadernación, almacenamiento y distribución tanto de obras o trabajos aislados como de publicaciones periódicas.

Del conocimiento generado en las instituciones y entidades cubanas solamente una pequeña parte es publicada en revistas seriadas o especializadas accesibles desde la intranet cubana, quedando la gran mayoría de estos materiales accesibles exclusivamente a los miembros de las instituciones donde fueron concebidos. Como parte del desarrollo científico alcanzado en las escuelas y universidades cubanas se ha generado un gran volumen de conocimiento en muchos casos documentado en tesis, libros, ponencias, manuales, tutoriales y artículos. Es común que el resultado de una investigación científica se divulgue solamente en el ámbito local donde se desarrolla y no encuentre un espacio donde difundirse de forma tal que pueda aprovecharse el resultado de la misma en otros lugares donde existan condiciones para su utilización.

En el año 2011 en la Universidad de Ciencias Informáticas se realizaron un total de 1342 publicaciones de corte científico, pero de estas 93 se encuentran en revistas referenciadas y solamente 14 son referenciadas en las bases de datos de Web of Science o Scopus, según datos del Departamento de Investigaciones de la institución [1]. Lo expresado anteriormente trae consigo que en ocasiones se dupliquen esfuerzos, trabajando sobre un tema que ya fue objeto de una investigación anterior en otro lugar del país.

Otro elemento que atenta en contra de la visibilidad del conocimiento documentado en Cuba, lo constituye la dificultad a la que se enfrentan las personas para compartir en la intranet nacional materiales que no cuentan con la formalidad necesaria para ser publicados en los espacios existentes, pero que no dejan de constituir un aporte importante en determinadas ramas de la ciencia y/o la cultura.

Materiales que son el resultado de conferencias científicas, seminarios, talleres y otros espacios cotidianos en entornos académicos, no cuentan hoy con un espacio donde se puedan compartir y enriquecer con los criterios y sugerencias de los cibernautas, lo que conlleva a que el conocimiento no sea fácilmente accedido y que no se pueda compartir entre entidades y personas de diferentes lugares.

INTRODUCCIÓN

Otra problemática que se deriva de la inexistencia de un espacio centralizado donde compartir documentos digitales en la intranet nacional, es que las obras literarias cubanas no tienen presencia y posicionamiento en Internet, lo que dificulta su difusión dentro y fuera del país como parte importante de la cultura nacional.

En los últimos años se ha incrementado la tendencia en el uso de las redes sociales, no solamente con fines de ocio y esparcimiento, sino como un medio para agrupar comunidades virtuales de interés, permitiendo que personas geográficamente distantes puedan compartir criterios y conocimiento sobre la base de la colaboración. Las redes sociales facilitan la posibilidad de información y actualización permanentes, la oportunidad de diálogo como elemento generador de intercambio, disponiendo de los aportes de cualquier integrante de una comunidad virtual, lo que actualmente no es explotado para la difusión de los documentos y publicaciones cubanas como medio para llegar a una mayor cantidad de personas.

Sobre la base de los elementos expuestos anteriormente se formula el siguiente **problema a resolver**: ¿Cómo mejorar la publicación y disponibilidad de documentos digitales en la intranet cubana?

Para darle solución a la problemática planteada se determinó como **objeto de estudio** los procesos de publicación y recuperación de documentos digitales, siendo el **campo de acción** los procesos de publicación y recuperación de documentos digitales en aplicaciones web y se ha definido como **objetivo general** del presente trabajo de diploma: Desarrollar una plataforma colaborativa mediante la informatización de los procesos de divulgación y recuperación de documentos digitales para mejorar su publicación y disponibilidad en la intranet cubana.

El objetivo general se deriva en los siguientes objetivos específicos:

- Caracterizar los fundamentos teórico-metodológicos que enmarcan los sistemas de publicación de documentos digitales.
- Proponer una plataforma colaborativa para la publicación de documentos digitales para la intranet cubana.
- Validar el correcto funcionamiento de la plataforma colaborativa desarrollada para la publicación en la intranet cubana de documentos digitales.

Teniendo en cuenta lo anterior, se plantea la siguiente **idea a defender**: El desarrollo una plataforma colaborativa para la intranet cubana mejorará la publicación y disponibilidad de documentos digitales.

El presente trabajo está compuesto por tres capítulos, los cuales guiarán la comprensión del proceso llevado a cabo para dar cumplimiento a los objetivos que fueron planteados. Se describen en orden

INTRODUCCIÓN

secuencial y lógico todo el proceso de investigación llevado a cabo para el desarrollo de una plataforma colaborativa para la publicación y recuperación de documentos digitales en la intranet cubana. A continuación se muestra una breve descripción de cada capítulo.

Capítulo I. Fundamentación teórica y estado del arte:

Se describe el marco teórico conceptual donde se definen los principales conceptos asociados al objeto de estudio. Se hace un estudio del estado del arte, así como un análisis de las principales herramientas, tecnologías y lenguajes usados en el mundo para la construcción de sistemas similares y la justificación de la selección de los que serán empleados en el desarrollo del sistema propuesto.

Capítulo II. Análisis y diseño:

Se realiza un análisis de los procesos existentes que permitan derivar requerimientos funcionales y no funcionales. Se realiza también la descripción de los casos de uso que serán implementados y se muestran estos en el diagrama de casos de uso del sistema. Se obtiene: el diagrama de clases del diseño, los diagramas de secuencia y la descripción de las clases; a partir de los casos de uso del sistema. Además se describe la distribución del sistema propuesto a través del diagrama de despliegue.

Capítulo III Implementación y prueba:

Se muestra el diagrama de componentes, el cual facilita la comprensión del flujo de trabajo de implementación. También se muestran las partes principales de la interfaz de usuario de la aplicación, así como los casos de pruebas del sistema y sus resultados.

Entre los aportes prácticos que se esperan lograr con el presente trabajo se encuentran la obtención de una plataforma colaborativa que permita la publicación de documentos digitales, que brinde a los usuarios la posibilidad de divulgar todo tipo de documentos digitales. También se ofrecerá un entorno colaborativo que permita emitir comentarios y criterios sobre los materiales disponibles para favorecer el intercambio de conocimientos y opiniones. El sistema obtenido aprovechará las potencialidades que brindan las redes sociales para la difusión de información en Internet mediante la recomendación de documentos. También se facilitará la ejecución de búsquedas básicas y avanzadas sobre la base de datos que contiene los documentos publicados en el sistema para facilitar el acceso y la recuperación de información.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El estudio de las principales aplicaciones que se encargan de la publicación y recuperación de documentos digitales posibilita entender cómo han evolucionado estos sistemas y saber cuáles son las tendencias actuales en esta área. El análisis de estas aplicaciones en el mundo, Cuba y en la Universidad de las Ciencias Informáticas, sirve como base para comprender el estado y desarrollo de las mismas. Para un buen entendimiento del desarrollo de aplicaciones web que gestionan documentos digitales, es necesario dominar las herramientas y tecnologías en las que se basan, así como hacer un análisis crítico y comparativo sobre las tecnologías que pueden ser utilizadas.

1.1. Descripción general del objeto de estudio

Para un mejor entendimiento y lograr enmarcar el objeto de estudio de la investigación, es de vital importancia conocer en qué consisten los procesos de publicación y recuperación de documentos digitales.

En esta nueva era informativa parece ser el documento digital uno de los frutos de la interacción entre hombre y máquina condicionado por dos elementos importantes: el aumento de necesidades informativas y el uso de tecnologías cada vez más avanzadas en las que el consumidor es cada vez más protagonista. Por lo que se hace necesario se hace necesario definir qué es lo que se entiende por el término documento digital.

Documento digital: es la objetivación de un mensaje en un soporte físico transmisible y transformable en el espacio y en el tiempo a fin de remediar una necesidad informativa y constituirse en fuente de información para obtener nuevo conocimiento o como todo mensaje icónico incorporado a un soporte permanente y empleado con una finalidad informativa [2].

En el caso de los documentos digitales la información se caracteriza estar expresada en ceros y unos lo que potencia la facilidad para ser reproducida, transmitida y almacenada: facilidad que superaría con mucho cualquier forma de reproducción, transmisión y almacenamiento de información analógica [3].

Es importante señalar que la **disponibilidad** es la cualidad que posee un recurso digital al estar disponible siempre que se necesite acceder a él, en el contexto del presente trabajo está referido a los documentos digitales.

En cuanto al proceso de publicación digital o electrónica se encuentran varias definiciones que son de utilidad para alcanzar una mejor comprensión del término, estas se enuncian a continuación.

Capítulo #1 – Fundamentación teórica.

El proceso de **publicación digital** es aquel que se inicia en formato electrónico y se realiza mediante programas y computadoras en aplicaciones web. En este tipo de publicaciones el tránsito no implica sólo el cambio de portador y de lenguaje para elaborarla, sino que a partir de las posibilidades que ofrecen los programas para aplicaciones y la creatividad de sus productores, se requiere del desarrollo de un nuevo producto, con una estructura coherente y estética, capaz de transmitir la hipertextualidad, propia de la mente humana a los escritos [2].

La **publicación digital** es la transferencia pura y la diseminación electrónica de información mediante redes, y que incluye todos los métodos asociados a la producción de documentos con el uso de computadoras, aunque a veces, se describe como la aplicación de la computación al proceso de edición de la publicación tradicional. El término se refiere más precisamente al almacenamiento y la distribución de información mediante el uso de la comunicación electrónica [4].

La **publicación electrónica** presenta un libro, revista, folleto, catálogo o cualquier otro material en forma digital, igual a la forma impresa, pero con características propias de la digitalización: espacio mínimo, portabilidad, material no alterable por el paso del tiempo, facilidad y economía de distribución, entre otras. Una publicación digital es igual a cualquier otra impresa, pero ella puede ser leída sin necesidad de imprimirla [5].

Partiendo de las definiciones anteriores se puede concluir que el proceso de **publicación de documentos digitales** es: la publicación basada en el conjunto de procedimientos que permiten cumplir el ciclo completo del documento en forma electrónica o digital. Aprovecha las ventajas de la digitalización: espacio mínimo, portabilidad, material no alterable por el paso del tiempo, facilidad y economía de distribución. Se refiere al almacenamiento y la distribución de los documentos de forma digital mediante el uso de la comunicación electrónica.

Definiéndose como proceso de **recuperación** de información el conjunto de tareas mediante las cuales el usuario localiza, identifica, selecciona y accede a los recursos de información que le son útiles [6].

La operación de recuperación de información se divide en tres subtareas [7]:

1. Indexación, cuya finalidad es obtener una representación de los documentos que permita, de forma eficiente, su almacenamiento y comparación con las consultas.
2. Procesamiento de la consulta, para adecuarla a la representación interna de los documentos.
3. Recuperación y ranking de documentos a partir de la comparación entre las representaciones internas de consulta y documentos.

1.2. Sistemas similares

Para determinar las características que debe cumplir la aplicación a realizar es necesario hacer un análisis sobre los sistemas que existen y las tendencias actuales que se emplean en situaciones similares tanto en el ámbito internacional, nacional o local. En el capítulo se abordan los sistemas: Google Books, ComparteLibros, LibrosGratis, la Biblioteca Digital de la Biblioteca Nacional José Martí, y la aplicación web Sunshine existente en la Universidad de las Ciencias Informáticas (UCI), además del buscador Orión, los cuales tienen puntos comunes con el sistema a desarrollar. A continuación se hará una breve descripción de cada sistema y sus limitaciones en cuanto a su utilización, con el objetivo de estudiar sus ventajas y desventajas de forma tal, que contribuya a crear una nueva aplicación ajustable a las necesidades.

1.2.1. Google Books

Es una aplicación web dedicada principalmente a la búsqueda especializada de libros digitales en Internet que funciona como una extensión del buscador Google. También permite crear y gestionar bibliotecas, compartir libros con amigos y ver qué están leyendo. La interfaz de la aplicación es similar a la del buscador Google con la adición de ejemplares de libros digitales divididos por tópicos lo que la hace amigable al usuario.

Según la administración del sitio¹, el mismo posee una colección de ejemplares que supera los 3 000 000 de libros digitales que pueden ser adquiridos de forma gratuita, también se pueden encontrar más de 300 000 libros que están protegidos por el derecho de autor que se indica en que sitio pueden ser comprados. Las búsquedas que se realizan pueden ser básicas pero también es relevante el nivel de detalle para filtrar los resultados que se alcanza en las búsquedas avanzadas dado por la cantidad de parámetros a tener en cuenta en la misma.

El sistema permite acceder de forma rápida a la información de un libro determinado mostrando un resumen con los datos relevantes del mismo, además permite comentar los ejemplares así como añadirlos a la biblioteca personal de cada usuario. También cuenta con un programa de afiliación para editores y autores así como un proyecto para bibliotecas, ambos encaminados a digitalizar los libros para que se encuentren accesibles en Internet.

Entre las principales limitaciones que se evidencian en el sistema se deben mencionar que se limita a archivos en formato PDF y solo permite el trabajo con libros, lo que excluye otros tipos de publicaciones importantes y los que no son gratuitos solo pueden ser adquiridos en sitios que no brindan este servicio para Cuba. Para contar con una biblioteca personal es necesario tener una cuenta en el servidor de

¹ Viernes 9 de noviembre de 2012

Capítulo #1 – Fundamentación teórica.

correos de Gmail lo que limita al usuario, además solo permite la integración con la red social Google+. Como obstáculo para los desarrolladores también se puede enunciar que el código fuente de la aplicación no está disponible ni se cuentan con detalles de su composición, así como de las tecnologías y herramientas empleadas en su construcción o funcionamiento.

1.2.2. ComparteLibros

En el ámbito internacional existe la aplicación web: ComparteLibros desarrollada en España, con el objetivo de crear una comunidad que permita compartir información sobre libros y autores, así como recomendar libros. Esta aplicación realiza procesos como: la publicación y descarga de libros, escribir comentarios de un determinado libro, compartir un libro en redes sociales como Facebook, posee chat y foro, permite realizar búsquedas utilizando diferentes criterios como autor y título, listar todas las publicaciones de un determinado autor. Todo esto permite el acceso fácil y rápido desde cualquier lugar y el intercambio de opiniones y libros entre usuarios de todo el mundo.

A pesar de lo ventajoso de este producto y de poseer 39051 libros de 13171 autores hasta la fecha de hoy², tiene sus deficiencias en cuanto a: no poseer materiales de corte científico como tesis, ponencias, manuales, tutoriales y artículos. No permite realizar búsquedas avanzadas lo que dificulta la recuperación de un determinado ejemplar. Además su utilización implica riesgos relacionados con el respeto al derecho de autor y a la propiedad intelectual, ya que como el sistema no está alojado en el país, sus propietarios no están obligados a respetar las leyes cubanas en lo referente a este tema.

1.2.3. LibrosGratis

Otra aplicación web que se puede encontrar internacionalmente es LibrosGratis. Esta es una plataforma para el intercambio de libros digitales en español donde es posible descargar libros gratis una vez que se es miembro de esta comunidad. Adquirir libros gratis en español e inglés del catálogo de este espacio es fácil ya que posee en la fecha actual³ más de 30000 libros y más de 14000 autores. Esta aplicación en su página principal tiene etiquetados para acceder de forma rápida a: los últimos libros, los más destacados, los de mayor popularidad y además permite realizar búsquedas. También posee todos los libros organizados por categorías de forma tal que se puede encontrar el ejemplar que se busca de manera más eficiente.

Independientemente de las ventajas que pueda propiciar esta plataforma, se ve limitada para realizar diversas funciones como son:

² Lunes 12 de noviembre de 2012.

³ Martes 13 de noviembre de 2012, según la administración del sitio.

Capítulo #1 – Fundamentación teórica.

A pesar de su organización, las búsquedas que se pueden realizar se ven limitadas a obtener un resultado según el criterio definido por la aplicación, ya que no posee filtros para su ejecución. El tipo de contenido que maneja se ve limitado a libros solamente lo que excluye otro tipo de documentos. No se almacena información con los detalles de los libros almacenados en el sistema por lo que el usuario recibe toda la información del libro hasta que lo descarga e incluso la ausencia de datos correspondientes a los libros impide la ejecución de búsquedas avanzadas, ya que no se cuentan con aspectos para el filtrado de los resultados. Su empleo implica riesgos relacionados con el respeto al derecho de autor y a la propiedad intelectual, ya que el sistema no está alojado en Cuba, sus propietarios no están obligados a respetar las leyes cubanas.

1.2.4. Biblioteca Digital de la Biblioteca Nacional José Martí

En la Intranet cubana no existe un espacio que permita la gestión de publicaciones digitales, un intento incipiente de desarrollar esto se encuentra en la Biblioteca Digital de la Biblioteca Nacional José Martí (BDBNJM) donde se encuentran publicadas⁴ solamente 461 obras de 71 autores. Esta surge en el año 2000 con el propósito de reunir la obra escrita de autores cubanos entre los siglos XVII-XIX. Esta aplicación reúne una serie de obras, la mayoría de estas de corte artístico literario. Este sitio web representa un espacio ideal para investigadores, estudiantes, profesores y lectores amantes de la literatura cubana.

Independientemente de la facilidad de información que provee este portal, posee deficiencias en cuanto a: no permite realizar búsquedas por lo que el usuario debe conocer previamente donde se encuentra lo que está buscando y para acceder a las obras es preciso dominar quien es el autor de la que se desea obtener. Tampoco tiene documentos de corte científico, como tesis, manuales, artículos entre otros. Además no permite a los usuarios dar opiniones acerca de algún libro, ni publicar y compartir libros que no se encuentran en este espacio con otras personas.

1.2.5. Sunshine

En la Universidad de las Ciencias Informáticas existe una plataforma para la gestión de materiales digitales que se encuentra en fase de prueba: Sunshine actualmente cuenta con más de 3600 documentos. Esta aplicación fue desarrollada con el objetivo de crear una biblioteca digital centralizada que sea auto-gestionada por sus propios usuarios. Actualmente⁵ esta aplicación solo cuenta con dos funcionalidades: realizar búsquedas y consultar los últimos documentos publicados.

⁴ Miércoles 14 de noviembre de 2012, según la administración del sitio.

⁵ Viernes 16 de noviembre de 2012, según la administración del sitio.

Capítulo #1 – Fundamentación teórica.

Esta aplicación tiene algunas desventajas: solo soporta el formato PDF en sus archivos ya la recuperación de los mismos resulta algo complejo porque los resultados obtenidos no pueden ser filtrados ni se pueden ejecutar búsquedas avanzadas, además no brinda un entorno que permita la interacción entre los usuarios que visitan el sistema ya que no permite compartir los documentos digitales en las redes sociales, ni escribir comentarios acerca de un determinado documento, imposibilitando el intercambio de opiniones. Debido a que se encuentra en fase de pruebas y no es un producto terminado, por lo que no es estable, es común que ocurran errores en tiempo de ejecución. Además esta aplicación se encuentra desarrollada en lenguaje Python utilizando el marco de trabajo Web2py, lo que dificultaría su integración con sistemas que utilicen Symfony que es el marco de trabajo definido por el Centro de Ideoinformática para el desarrollo de aplicaciones web.

1.2.6. Orión

El motor de búsqueda Orión se encuentra actualmente desarrollándose en la Universidad de las Ciencias Informáticas (UCI). Este tiene como objetivo general brindar servicios para la recuperación eficiente de información en fuentes de datos estructuradas y no estructuradas. Orión permite realizar búsquedas generales en los documentos publicados y también la búsqueda avanzada, se puede filtrar los resultados por idioma, por tipos de documentos (pdf, html, entre otros), así como, opciones para buscar en cualquier parte de los documentos o en el título, palabras claves, entre otros.

Dispone también de una capa de servicios web que constituye una alternativa para los usuarios que deseen consumir los servicios del buscador sin acceder necesariamente a la página principal. Permite realizar consultas utilizando el correo electrónico y conocer datos de webmetría de los sitios recolectados que pueden ser muy útiles para los administradores de dichos portales.

Entre las principales tecnologías para el apoyo del desarrollo web que utiliza Orión se encuentran las siguientes:

Para la implementación de sus interfaces y la lógica de obtención de datos recogidos en la web se utiliza el marco de trabajo de desarrollo Symfony 2.0.

Para el almacenamiento en base de datos de reportes de consultas, directorios y otros datos de interés para la visualización de información utiliza el sistema gestor de base de datos MongoDB.

Este buscador incluye en sus resultados varios tipos de archivos entre ellos los documentos digitales, pero no brinda un entorno colaborativo para la interacción de los usuarios con los mismos. Además como es solamente un buscador y no una plataforma para la interacción con documentos digitales, está limitado en el trabajo con los mismos. No permite compartir contenidos en las redes sociales ni expresar criterios ni comentarios sobre los mismos.

Capítulo #1 – Fundamentación teórica.

A partir del estudio realizado se identificaron un grupo de características que permitieron realizar una evaluación homogénea a los distintos sistemas analizados y determinar si alguno constituye una posible solución parcial o total a la problemática identificada.

El estudio realizado se muestra en la tabla 1.

	Google Books	Compart e Libros	Libros Gratis	BDBNJ M	Sunshine	Orión
Publicación de documentos digitales	Si	Si	Si	No	Si	No
Formatos soportados	PDF	PDF	PDF, ppt, doc, xls	No	PDF	Varios
Tipos de documentos	Libros	Libros	Libros	Libros	Libros, publicaciones seriadas, tutoriales	Libros, publicaciones seriadas, tutoriales
Opción de comentarios	Si	Si	Si	No	No	No
Compartir documentos en redes sociales	Si(Solo en Google+)	Si	Si	No	No	No
Ejecución de búsqueda básica	Si	Si	Si	No	Si	Si
Ejecución de búsqueda avanzada	Si	No	No	No	No	Si
Mostrar información de documento	Si	Si	No	Si	Si	Limitada
Acceso al código fuente y posibilidad de modificación	No	No	No	No	Si	Si

Capítulo #1 – Fundamentación teórica.

Riesgo de soberanía tecnológica	Alto	Alto	Alto	Ninguno	Ninguno	Ninguno
---------------------------------	------	------	------	---------	---------	---------

Tabla 1. Características presentes en los sistemas estudiados.

Las alternativas existentes en la actualidad en Internet para la publicación de documentos digitales, suponen varios riesgos y dificultades para Cuba. Algunos sistemas como Google Books han restringido la utilización de sus servicios a países como Cuba y en otros casos como ComparteLibros y LibrosGratis suponen un riesgo de soberanía tecnológica y de propiedad intelectual, ya que no se tienen garantías ante la posibilidad de piratería de las obras así como que en algún momento estos sistemas restrinjan el acceso a Cuba.

Mediante el análisis anteriormente expuesto se pudo concluir que resulta necesario desarrollar una plataforma colaborativa que pueda funcionar de forma autónoma pero que a su vez pueda ser integrada al motor de búsqueda Orión, de forma tal que pueda complementar el funcionamiento del mismo. La plataforma que se propone desarrollar debe ser capaz de cumplir con las características que se definieron para analizar los sistemas estudiados.

Resulta de gran importancia solventar las deficiencias que se detectaron en el ámbito nacional y local para brindar una solución efectiva a la problemática existente. Se pueden aprovechar las buenas prácticas identificadas en los sistemas estudiados, principalmente los elementos relacionados con las búsquedas avanzadas (muy potentes en Google Books). También resulta vital tener en cuenta las tendencias actuales encaminadas hacia la integración con las redes sociales, así como la posibilidad de ofrecer a los usuarios un ambiente colaborativo, elementos estos que se pueden apreciar a nivel internacional.

1.3. Metodologías de desarrollo

Existen varias formas de desarrollar un sistema pero si no se aplica una metodología de desarrollo coherente y organizada se corre el riesgo de no obtener como resultado final una aplicación de calidad y con ello que no se satisfagan las necesidades del cliente. También puede ocurrir que no se tenga un correcto control sobre los cambios que se soliciten, provocando atrasos dañinos en el desarrollo del proyecto. Es por esto que los desarrolladores necesitan una guía para crear un producto con calidad y eficiencia.

Debido a que el grupo de desarrollo es pequeño, el tiempo de la construcción de la plataforma colaborativa es poco y teniendo en cuenta los objetivos anteriormente trazados, se ha decidido utilizar una metodología de desarrollo ágil ya que es lo más adecuado teniendo en cuenta las circunstancias.

1.3.1. AUP (Agile Unified Process)

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad [8].

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos [8].

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP [8].

1.3.2. SCRUM

Más que una metodología de desarrollo de *software*, es una forma de auto-gestión de los equipos de programadores. Un grupo de programadores deciden cómo hacer sus tareas y cuánto van a tardar en ello. SCRUM ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro. Permite además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo [9].

SCRUM se basa en un enfoque iterativo, donde cada iteración se denomina Sprint. El principio básico es que es muy difícil contar desde el principio con un catálogo completo de funcionalidades, ya que los requisitos van surgiendo conforme el propietario de la aplicación y los usuarios de la misma van haciendo sucesivas aportaciones. Así pues, SCRUM plantea el desarrollo de sucesivas versiones ampliadas, todas ellas plenamente usables y evaluables por el usuario. SCRUM es, además, una metodología especialmente indicada para pequeños equipos de desarrollo y se orienta a la entrega rápida de resultados y una alta flexibilidad [9].

Esta metodología ofrece ventajas como [9]:

Capítulo #1 – Fundamentación teórica.

- Permite a las organizaciones eliminar los impedimentos clásicos en el desarrollo de los proyectos, aumentando la satisfacción de los clientes mediante la realización de entregas frecuentes de resultados tangibles e integrándolos activamente en el ciclo de desarrollo, lo cual proporciona además una mayor adaptación y adecuación a sus necesidades.
- Potencia la formación de equipos de trabajo autosuficiente y multidisciplinarios, reduciendo la carga de gestión y proporcionando a los miembros del equipo un entorno amigable y productivo para desarrollar sus habilidades al máximo. Este entorno proporciona además mayor calidad de vida a los trabajadores y mejora drásticamente la moral en las organizaciones.
- Se centra en el producto y las personas, y hace especial hincapié en la eliminación preactiva de todas las trabas e impedimentos que surjan durante el desarrollo. Así pues, permite a muchas organizaciones alcanzar el llamado “Efecto Toyota”: cuatro veces la productividad media del sector, con doce veces la calidad.
- SCRUM es iterativo e incremental. Eso quiere decir que se libera código (o producto) frecuentemente, y cada liberación representa un incremento sobre la anterior.
- El producto que funciona es la única medida del avance del proyecto.
- Las entregas son productos potencialmente utilizables y/o comercializables. Eso significa que nada de presentación y nada de planos o documentos que describen lo que hará la aplicación.

1.3.3. XP

Metodología de desarrollo de *software* exitosa utilizada para proyectos de corto plazo y equipos pequeños; ha generado gran interés por su creciente número de casos de éxito en la industria. Metodología ágil que requiere gran disciplina. Un proceso ligero de bajo riesgo, flexible, predecible, científico y divertido de desarrollar *software*. Su utilidad se mide en valores como la simplicidad, comunicación, realimentación y coraje.

La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Las entregas son frecuentes, y existe una refactorización continua, lo que permite mejorar el diseño cada vez que se le añade una nueva funcionalidad. Sustenta la propiedad colectiva, la programación en parejas y pequeños encuentros diarios donde cada uno de los integrantes del equipo cuenta que ha hecho, que problemas tiene, y que hará más adelante [10].

¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.

Capítulo #1 – Fundamentación teórica.

- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o la etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y proporcionando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

1.3.4. OpenUP

La metodología de desarrollo OpenUp se define como un proceso modelo y extensible, dirigido a gestión y desarrollo de proyectos de *software* basados en desarrollo iterativo, ágil e incremental apropiado para proyectos pequeños y de bajos recursos; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo. Esta posee un proceso iterativo del desarrollo del *software* que es mínimo, completo, y extensible. Es un proceso para equipos de desarrollo pequeños y que le dan valor a la colaboración y a las necesidades de los usuarios que utilizarán la aplicación.

Es mínimo, porque solo incluye el contenido del proceso fundamental; completo, ya que logra ser manifestado como proceso entero para construir un sistema; y es extensible, porque puede ser utilizado como base para agregar o adaptar según las necesidades. Es un proceso que apoya el desarrollo iterativo, ágil, e incremental y es aplicable a un amplio sistema de plataformas y de usos del desarrollo.

OpenUP estructura el ciclo de vida de un proyecto en cuatro fases: concepción, elaboración, construcción y transición. El ciclo de vida del proyecto provee a los interesados un mecanismo de supervisión y dirección para controlar los fundamentos del proyecto, su ámbito, la exposición a los riesgos, el aumento de valor y otros aspectos [11].

- **Concepción:** Primera de las 4 fases en el proyecto del ciclo de vida, acerca del entendimiento del propósito y objetivos y obteniendo suficiente información para confirmar que el proyecto debe hacer. El objetivo de ésta fase es capturar las necesidades de los usuarios finales en los objetivos del ciclo de vida para el proyecto.
- **Elaboración:** Es el segundo de las 4 fases del ciclo de vida del OpenUP donde se trata los riesgos significativos para la arquitectura. El propósito de esta fase es establecer la base la elaboración de la arquitectura del sistema.

Capítulo #1 – Fundamentación teórica.

- **Construcción:** Esta fase está enfocada al diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de esta fase es completar el desarrollo del sistema basado en la Arquitectura definida.
- **Transición:** Es la última fase, cuyo propósito es asegurar que el sistema es entregado a los usuarios, y evalúa la funcionalidad y performance del último entregable de la fase de construcción.

La metodología de desarrollo OpenUp tiene como principio fundamental colaborar para sincronizar intereses y compartir conocimiento. Este principio promueve prácticas que impulsan un ambiente de equipo saludable, facilitan la colaboración y desarrollan un conocimiento compartido del proyecto. [4]

Partiendo de que los principios de la metodología de desarrollo OpenUp son: colaborar para alinear intereses y para compartir conocimiento, balancear las prioridades para maximizar las necesidades de los usuarios finales, centrado en la arquitectura y desarrollo iterativo, y que estos responden a las características que presenta el grupo de desarrollo y el trabajo que se desea realizar, se decide utilizar OpenUp como metodología de desarrollo de *software*. Esta metodología provee beneficios como: disminuir las probabilidades de fracaso e incrementar las probabilidades de éxito ya que es apropiado para proyectos pequeños y de bajos recursos, permite detectar errores tempranos a través de un ciclo iterativo, evita la elaboración de documentación, diagramas e iteraciones innecesarios y por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas [11].

1.4. Lenguaje de modelado

Lenguaje Unificado de Modelado (UML) 2 se centra en la representación gráfica de un sistema, capturando las partes esenciales del mismo. Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- **Visualizar:** Permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- **Especificar:** Permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado y pueden ser utilizados para su futura revisión.

Capítulo #1 – Fundamentación teórica.

El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático, desde el análisis con los casos de uso, el diseño con los diagramas de clases, hasta la implementación y configuración con los diagramas de despliegue.

UML aporta beneficios al desarrollo de una aplicación, ya que permite realizar una representación gráfica de la misma con un gran rigor en la especificación de sus componentes, lo que proporciona un mayor entendimiento de sus características y facilita su proceso de desarrollo.

El UML sirve para visualizar gráficamente sistemas complejos, tanto en el diseño del *software* como de la arquitectura hardware donde se ejecutan, siendo esta la razón fundamental por la que se decide utilizar UML como lenguaje para la representación gráfica y documentación de los artefactos generados en el desarrollo de esta aplicación. [12]

1.5. Lenguajes de codificación

Un lenguaje de codificación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y *software* existentes. Actualmente existen diversos lenguajes para desarrollar en la web, ya que, con el avance de las tecnologías surgieron lenguajes que permiten interactuar con los usuarios haciendo uso de sistemas de base de datos.

Para el desarrollo de la plataforma colaborativa para la gestión de publicaciones digitales se decidió utilizar los lenguajes del lado del cliente HTML, CSS y JavaScript y como lenguaje del lado del servidor PHP. A continuación se describen las características de cada uno y los beneficios que aportan a la aplicación.

1.5.1. Lenguaje de Marcado de Hipertexto HTML 4

El lenguaje HTML: siglas en inglés (HyperText Markup Language) es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium (<http://www.w3.org/>), más conocido como W3C. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo.

El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje

Capítulo #1 – Fundamentación teórica.

utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica. [13]

1.5.2. Hojas de Estilo en Cascada 3

Una hoja de estilo en cascada CSS (Cascading Style Sheets) es un lenguaje creado por el W3C que permite a los desarrolladores controlar los aspectos de estilo y formato de múltiples páginas HTML simultáneamente, permitiendo separar el contenido de presentación en todo momento.

El empleo de hojas de estilo CSS es una práctica que aporta una serie de beneficios como:

- Aumento de la legibilidad y reducción del peso de las páginas Web.
- Mejora del mantenimiento y actualización de los sitios Web.
- Mejora de la accesibilidad: se pueden definir hojas de estilo locales en función de las necesidades o preferencias del usuario.
- Versatilidad: Se ofrecen diferentes hojas de estilo para los diferentes tipos de medio existentes (sintetizadores de voz, dispositivos braille, pantallas de computador a color, televisión).

El lenguaje CSS da la posibilidad de tener el control del diseño en la aplicación, ya que como se ha dicho, es muy útil para separar el contenido del diseño, además permite personalizar la aplicación. [14]

1.5.3. JavaScript 1.8

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

JavaScript es seguro y fiable ya que los scripts tienen capacidades limitadas por razones de seguridad, es ejecutado en el cliente por lo que el servidor no es utilizado más de lo debido, aunque tiene la desventaja que el código del script debe descargarse completamente para poder ejecutarlo.

Este lenguaje brinda la posibilidad de establecer una conexión asíncrona con el servidor; lo que garantiza que no haya una sobrecarga de peticiones al mismo y el aumento de la velocidad de respuesta. Además, con JavaScript se puede tener el control total de las animaciones y la manipulación de los componentes en la página web. [15]

1.5.4. PHP 5.3

PHP: (Hypertext Preprocessor) es un lenguaje de código abierto muy popular, especialmente adecuado para desarrollo Web y que puede ser incrustado en HTML. Con PHP se puede procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.

Las características que distinguen a este lenguaje y que le hacen uno de los más usados son:

Puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix, Microsoft Windows, Mac OS X, RISC OS, entre otros.

Soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros.

Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.

Una de las ventajas que brinda el lenguaje PHP al utilizarlo en la plataforma colaborativa para la gestión de publicaciones digitales es la posibilidad de programar orientado a objetos, lo cual proporciona mucha claridad al desarrollo. Es uno de los lenguajes de programación web que más documentación tiene, dispone de un sitio oficial que brinda toda la documentación que los desarrolladores necesiten, además existe una gran comunidad de desarrollo que aporta sus conocimientos. [16]

1.6. Marco de trabajo

En el mundo de desarrollo de aplicaciones informáticas el uso de los marcos de trabajo se vuelve inminente, ya que estas estructuras conceptuales y tecnológicas de soporte definido, constituyen la base con la cual otro proyecto de *software* puede ser fácilmente organizado y desarrollado. [17]

El término marco de trabajo se refiere a una estructura de *software* compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, se puede considerar como una aplicación genérica incompleta y configurable a la que se puede añadir las últimas piezas para construir una aplicación concreta. [18]

Los objetivos principales que persigue un marco de trabajo son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. [18]

Capítulo #1 – Fundamentación teórica.

Un marco de trabajo de desarrollo web, por tanto, se puede definir como un conjunto de componentes que forman un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web.

Se tuvo en cuenta la posible integración de la plataforma que se desea desarrollar con el motor de búsqueda Orión, por lo que se decidió que en el desarrollo de la misma se utilizará como marco de trabajo de desarrollo web para el apoyo a la implementación Symfony, en su versión 2.0, además de que es el marco de trabajo de desarrollo para el lenguaje PHP que se encuentra aprobado por el centro CIDI al cual pertenece el presente trabajo de diploma. A continuación se enuncian las posibilidades que esta brinda basándonos en sus principales características.

1.6.1 Symfony 2.0

Es un sistema muy versátil ya que permite la integración plena entre los componentes, librerías externas y bundles, lo que facilita el desarrollo de aplicaciones que utilicen gran variedad de estos elementos.

Utiliza buenas prácticas en el desarrollo de aplicaciones web obtenidas de otros marcos de trabajo utilizando características y estilos que definen lo mejor de cada uno de ellos.

Es muy flexible ya que en el desarrollo de cada uno de los componentes de su arquitectura permite utilizar varios lenguajes y tecnologías por lo que no existe un modelo único sino un conjunto de opciones.

Optimiza el rendimiento ya que todo el código de las plantillas es compilado en PHP 5.3.X lo que permite utilizar las ventajas que brindan estas versiones y además utiliza un proxy inverso para disminuir el tiempo de respuesta de las peticiones.

Se encuentra basado en el patrón de diseño Modelo Vista Controlador (MVC) lo que permite establecer una separación entre la lógica del negocio, la lógica del servidor y la presentación de la aplicación web.

Automatiza el funcionamiento de las tareas comunes lo que le permite al desarrollador centrarse en las que son específicas de la aplicación que desarrolla.

1.7. Sistema Gestor de Base de Datos

Los sistemas de gestión de bases de datos (SGBD) son un tipo de *software* dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de estos sistemas es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. [19]

Por su parte, los almacenes de datos pueden ser relacionales y no relacionales, a estos últimos también se les conoce comúnmente como bases de datos No SQL. [19]

Capítulo #1 – Fundamentación teórica.

Las bases de datos relacionales cumplen con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer conexiones entre los datos que están guardados en tablas y a través de dichas conexiones relacionar los datos de ambas tablas; de ahí proviene su nombre: "Modelo Relacional". [19]

Por otro lado, los almacenes de datos no relacionales no proporcionan garantías ACID⁶. Normalmente no tienen esquemas fijos de tablas ni sentencias "join"⁷. Estos sistemas responden a las necesidades de escalabilidad horizontal que tienen cada vez más empresas.

Dentro de los sistemas de gestión de bases de datos relacionales usados a nivel internacional se encuentran: PostgreSQL, MySQL, ORACLE y SQL-Server.

Actualmente los almacenes de datos No SQL que más sobresalen son: MongoDB, Cassandra y BigTable, todos usados en aplicaciones que manejan grandes volúmenes de datos como son: Twitter, Facebook y Google respectivamente.

1.7.1. PostgreSQL 9

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales [19].

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando [19].

1.7.2. MongoDB 2.4

Un estudio enfocado en las base de datos no relacionales arroja que MongoDB constituye una solución escalable y de alto rendimiento de almacenes de datos No SQL.

De código abierto y escrito en C++, este sistema tiene orientado el almacenamiento de datos en documentos al estilo JSON con esquemas dinámicos, que ofrecen potencia y simplicidad. Se destaca por conservar los índices de todos los atributos y hacer mucho más flexible la agregación y procesamiento de datos [19].

Esto tiene ventajas y desventajas, sin embargo, existe un sesgo en el esquema, que puede hacer ciertas consultas un poco menos elegante que en el esquema relacional.

⁶ Conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.
⁷ Sentencia para relacionar tablas.

Capítulo #1 – Fundamentación teórica.

Las colecciones en MongoDB son análogas a las tablas en una base de datos relacional. Cada colección contiene varios documentos. Como se mencionó anteriormente estos documentos pueden ser grandes.

No existe ninguna declaración explícita de los campos de los documentos de una colección. Sin embargo, hay un diseño conceptual del esquema de lo que esos campos representarán y cómo los documentos de la colección estarán estructurados. [20]

MongoDB no requiere que en una colección todos los documentos tengan la misma estructura. Sin embargo, en la práctica, la mayoría de las colecciones son altamente homogéneas. [20]

Dada las características planteadas se selecciona MongoDB para el almacenamiento de los datos relacionados con la plataforma colaborativa. La selección de este sistema de almacenamiento de datos se justifica porque logra un mejor manejo de la información frente a un volumen considerable de datos, ya que es necesario manejar una gran cantidad de documentos y metadatos asociados a los mismos. Además de que el marco de trabajo a utilizar (Symfony2), posee soporte para MongoDB.

1.8. Servidor web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente. Estos generan una respuesta en cualquier lenguaje entendible por el cliente. Para la transmisión de todos estos datos suele utilizarse generalmente el protocolo HTTP, perteneciente a la capa de aplicación del modelo OSI. [21]

1.8.1 Apache 2

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. [21]

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la *Apache Software Foundation*. Presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Apache tiene amplia aceptación: desde 1996, Apache, es el servidor HTTP más usado. [21]

Este servidor web es redistribuido como parte de varios paquetes propietarios de *software*, incluyendo la base de datos Oracle y el IBM WebSphere Application Server. Mac OS X integra apache como parte de su propio servidor web y como soporte de su servidor de aplicaciones WebObjects. Apache es incluido con

Capítulo #1 – Fundamentación teórica.

Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux. Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. [21]

Algunos de los grandes sitios web del mundo están ejecutándose sobre Apache. La capa frontal (front end) del motor de búsqueda Google está basada en una versión modificada de Apache, denominada Google Web Server (GWS). La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original. [21]

Por las características anteriormente expuestas se selecciona Apache 2 como servidor web para la plataforma colaborativa, pues es un servidor web de código abierto para plataformas Unix (BSD, GNU/Linux) y viene en la instalación por defecto de muchos de estos sistemas operativos. Además de la necesidad de que todos los documentos y la información obtenida, pueda ser publicada de una forma segura y confiable en la plataforma colaborativa.

1.9. Herramientas

Las herramientas son productos que dan soporte a una tarea concreta dentro de las actividades de desarrollo de *software*. Dicho soporte consistirá en una serie de servicios, cada uno de los cuales automatiza una operación individual. Estas se pueden clasificar según los servicios que ofrece y/o la tarea a la que da soporte. A continuación se describen las herramientas que deben ser utilizadas en el entorno de desarrollo de la aplicación:

1.9.1. Herramienta CASE de modelado

Las herramientas CASE (por sus siglas en inglés: **C**omputer **A**ided **S**oftware **E**ngineering, o Ingeniería de Software Asistida por Computadora) son aquellas aplicaciones que brindan ayuda al desarrollo de *software*, disminuyendo su costo en términos de tiempo y de dinero. Garantizan una correcta documentación, generación de código y reutilización del *software*.

Visual Paradigm for UML 6.4 (VP-UML)

Es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite realizar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Una de sus características más importantes es que es multiplataforma.

Visual Paradigm es la herramienta CASE seleccionada para el modelado UML debido a que soporta el ciclo de vida completo del desarrollo de *software*, por lo que facilita la construcción de sistemas con

Capítulo #1 – Fundamentación teórica.

calidad y de forma rápida; permite la generación de código y de bases de datos. Además es multiplataforma, soporta ingeniería inversa, genera documentación del sistema en formato HTML y PDF. [22]

1.9.2. Netbeans 7.2

Esta herramienta permite desarrollar múltiples proyectos en disímiles lenguajes de programación como son: Symfony2, Java, PHP y otros. Posee una amplia documentación, facilidades de autocompletamiento y generación automática de código. Así como también incluye muchas librerías y pluing para varios lenguajes de programación. Se selecciona esta herramienta para el desarrollo de la plataforma colaborativa para la gestión de publicaciones digitales debido a que al utilizar esta herramienta para desarrollar un proyecto en Symfony2 obtienes los siguientes beneficios. [23]

Al crear el proyecto Symfony2 desde Netbeans:

No es necesario tener que configurar VirtualHosts en Apache, es decir, que permite centrarse en el desarrollo del proyecto.

Se pueden ejecutar comandos de la consola de Symfony2 desde Netbeans, como por ejemplo para la creación de: bundles, entidades y formularios. Además Netbeans facilita de manera interactiva una ayuda de cada comando, así como los parámetros de cada uno.

La pestaña Proyectos proporciona una visión global de la estructura de directorios del proyecto, permitiendo navegar fácilmente a través de ella desde el propio Netbeans, así como crear nuevos archivos.

El autocompletado brinda la sintaxis de la mayoría de las clases y métodos de Symfony 2, lo que incrementa notablemente la velocidad de programación.

1.10. Conclusiones

El estudio de los sistemas homólogos permitió determinar que ninguno soluciona totalmente la problemática planteada y se analizaron las potencialidades que pudiesen ser incorporadas a la solución, además se detectaron las tendencias actuales en el desarrollo de este tipo de *software*. Se definió como metodología de desarrollo OpenUp y entre los elementos que se decidieron emplear en el desarrollo del sistema se encuentran: el lenguaje de modelado UML 2, entre los lenguajes a utilizar se pueden encontrar HTML 4, JavaScript 1.8, CSS 3 y PHP 5.3, como marco de trabajo Symfony 2.0 y Apache 2 como servidor web, Netbeans 7.2 para la gestión de código fuente y Visual Paradigm for UML 6.4 (VP-UML) para el modelar el sistema.

Capítulo #2 – Análisis y diseño.

CAPÍTULO #2: ANÁLISIS Y DISEÑO DE LA PLATAFORMA COLABORATIVA PARA LA PUBLICACIÓN DE DOCUMENTOS DIGITALES

En todo desarrollo de sistemas de *software* es de suma importancia el seguir alguna especificación que permita a los desarrolladores el tener una disciplina que haga que todas las etapas sean no solo más coherentes sino también más formales. Se propone, al ser un *software* que puede tener aplicación dentro del contexto de un problema real, seguir un proceso de análisis y diseño que proporcione los cimientos bajo los cuales se va a desarrollar la aplicación. Un adecuado estudio del flujo actual de los procesos influye directamente en que la solución propuesta satisfaga las necesidades del cliente. Es por esto que en el presente capítulo se detallan los procesos de análisis y diseño según define la metodología utilizada.

2.1. Propuesta del sistema

La plataforma colaborativa para la gestión de documentos digitales es un sistema informático que integrará varias funcionalidades las cuales permiten a los usuarios la publicación y descarga de documentos digitales en varios formatos, además permitirá realizar búsquedas básicas y avanzadas. También posibilitará comentar acerca de los materiales publicados, así como compartir estos en las redes sociales y brindar su información asociada. Otras de las funcionalidades con las que contará esta aplicación son: la gestión y autenticación de los usuarios en el sistema garantizando la seguridad y confiabilidad de los documentos que se encontrarán en esta plataforma. También brindará opciones de envío de correo informando de lo que sucede en la aplicación con algún documento o categoría.

En la plataforma los usuarios serán los encargados de la gestión de usuarios y documentos a partir de privilegios que posean. Esta aplicación funcionará de manera autónoma aunque podrá ser integrada al motor de búsqueda Orión, con el objetivo de complementar sus funcionalidades de manera que se amplíen sus servicios para la publicación y búsqueda de información en la web. El sistema se basará en la estructura que define el marco de trabajo Symfony2.0, potenciado por el gestor de bases de datos MongoDB aprovechando las ventajas que brindan estos sistemas.

2.2. Modelo de dominio

Un **modelo del dominio** es una representación *visual* de las clases conceptuales u objetos del mundo real en un dominio de interés [24]. Este representa clases conceptuales del dominio del problema, conceptos del mundo real, no de los componentes de *software* [25].

Capítulo #2 – Análisis y diseño.

Debido a que los procesos del negocio no se encuentran bien definidos se decidió desarrollar un modelo de dominio que represente los conceptos relacionados con estos y las relaciones que se establecen entre ellos.

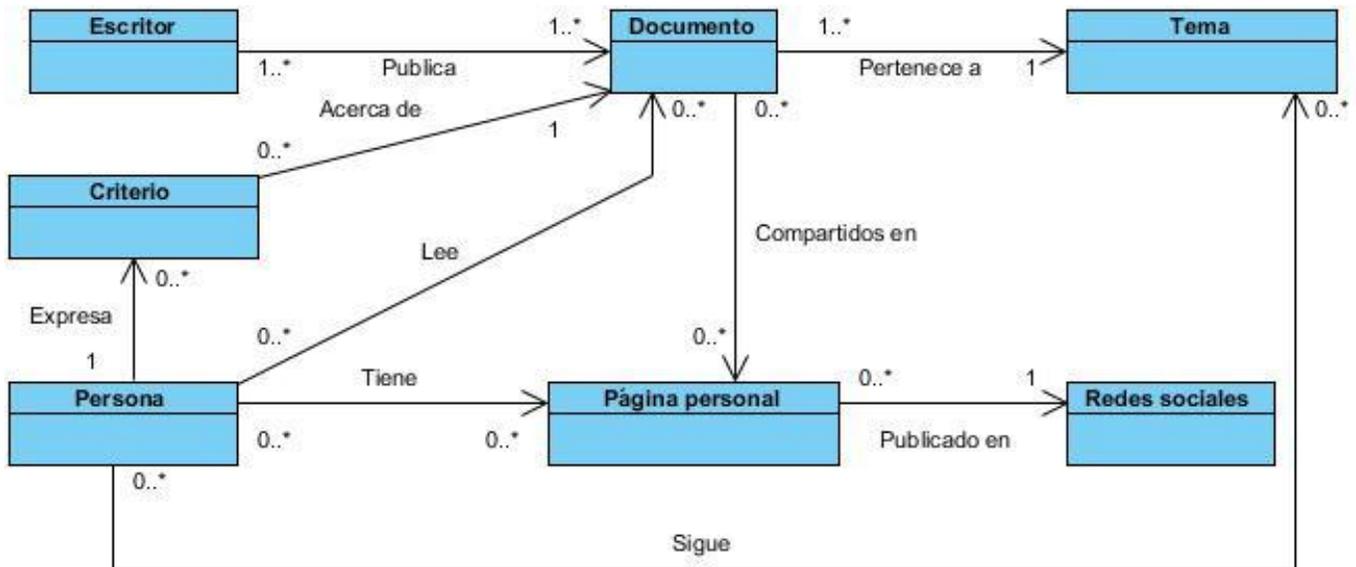


Ilustración 1. Diagrama de clases del modelo de dominio.

A continuación, se describen cada una de las clases del modelo de dominio.

Escritor: Es la persona que escribe y publica los documentos.

Documento: Comprende todo tipo de documentación digital puede ser un libro, un artículo, una revista, una ponencia, el resultado de una investigación o cualquier otro documento. Son publicados por un escritor y pertenecen a un tema determinado.

Persona: Lee los documentos y expresa criterios acerca de los documentos leídos, pueden tener páginas personales publicadas en las redes sociales.

Criterio: Son expresados por las personas, están relacionados con los documentos leídos, pueden ser un comentario, una puntuación o simplemente decir que le gustó un documento.

Tema: Comprende los temas a los que pertenecen los documentos, estos pueden ser seguidos, de acuerdo a sus intereses, por las personas.

Página personal: Pertenecen a las personas y están publicadas en las redes sociales, contienen un perfil con la información personal de cada usuario registrado en las redes sociales, además de información que este desea compartir con la comunidad, incluye documentos y otros materiales.

Redes Sociales: Plataformas existentes en Internet, cuyo propósito es facilitar la comunicación y el debate en la red. Contienen las páginas personales con la información de millones de usuarios.

Capítulo #2 – Análisis y diseño.

2.3. Relación de los requerimientos del sistema

Se define como requerimiento a la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. [24]

2.3.1. Técnicas de captura de requisitos

La captura de requisitos ayuda al equipo de desarrollo de un sistema a entender mejor el problema en cuya solución se trabajará. Este proceso incluye un conjunto de tareas que conducen a comprender, qué es lo que el cliente quiere y como interactuarán los usuarios finales del *software*. Por la complejidad que esto implica, el equipo de desarrollo empleó las siguientes técnicas que permitieron hacer este proceso de una forma más eficiente y precisa.

Tormenta de ideas: es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la acumulación de ideas e información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. Como técnica de captura de requisitos es sencilla de usar y de aplicar [26].

La técnica tormenta de ideas fue utilizada durante las reuniones del grupo de desarrollo, donde se determinaron un grupo de posibles funcionalidades que pudieran ser incorporadas a la aplicación que se desea desarrollar.

Sistemas existentes: consiste en analizar los sistemas ya desarrollados que estén relacionados con el sistema que va a ser construido. Analizándose las interfaces de usuarios y las entradas y salidas que el mismo produce [26].

El uso de la técnica sistemas existentes se hace evidente durante la realización del proceso de estudio de los sistemas homólogos, donde la mayor repercusión de la misma se encuentra reflejada en el análisis realizado al final de este estudio, ya que se determinaron funcionalidades que se incluirán en la aplicación que se desarrollará.

2.3.2. Requerimientos funcionales

RF1. Registrar usuario.

RF2. Modificar datos de usuario.

RF3. Eliminar usuario.

RF4. Autenticar usuario.

Capítulo #2 – Análisis y diseño.

RF5. Cerrar sesión de usuario.

RF6. Modificar permisos de usuario.

RF7. Bloquear usuario.

RF8. Proponer documento.

RF9. Modificar datos de documento.

RF10. Eliminar documento.

RF11. Descargar documento.

RF12. Comentar documento.

RF13. Eliminar comentario.

RF14. Compartir documento.

RF15. Votar documento.

RF16. Gustar documento.

RF17. Seguir documento.

RF18. Publicar documento.

RF19. Seguir tema.

RF20. Realizar búsquedas básicas.

RF21. Realizar búsquedas avanzadas.

RF22. Mostrar información de un documento.

RF23. Confirmar usuario.

RF24. Dejar de seguir documento.

RF25. Dejar de seguir tema.

RF26. Adicionar tema.

RF27. Eliminar tema.

RF28. Enviar reporte.

RF29. Buscar usuario.

Para ver las descripciones de los requisitos funcionales, consultar **Anexo 1**.

Capítulo #2 – Análisis y diseño.

2.3.3. Requerimientos no funcionales

Usabilidad

RnFU1: La interfaz de usuario debe diseñarse para una resolución de 1024x768.

RnFU2: Deben usarse colores preferentemente claros y similares a los del motor de búsqueda Orión.

Eficiencia

RnFE1: El sistema no debe tardar más de 8 segundos en procesar la petición del usuario.

Hardware⁸

RnFH1: El sistema necesita un servidor que posea 16 GB de Memoria RAM para ser desplegado.

RnFH2: El sistema necesita un servidor que posea un Procesador: Core i3 o superior para ser desplegado.

RnFH3: El sistema necesita un servidor que posea un Disco Duro: 500 GB para ser desplegado.

Seguridad

RnFS1: El sistema cuenta con conjunto de roles que hacen más segura la aplicación restringiendo los permisos de cada usuario.

RnFS2: La sesión de un usuario debe expirar después de estar inactivo por más de 10 minutos.

2.4. Modelo de casos de uso del sistema

En esta sección se definen los actores y se describen los casos de uso principales del sistema, quedando conformado el Diagrama de Casos de Uso del Sistema.

2.4.1. Definición de los actores del sistema

Los actores del sistema especifican un rol jugado por un usuario o cualquier otro sistema externo que interactúa con la aplicación.

A continuación se muestran los actores y la justificación que tienen en el sistema.

Actores del Sistema	Justificación
Usuario anónimo	El actor representa a los usuarios que

⁸ Estos requisitos pueden ser adaptados al entorno en que se despliegue el sistema, está concebido para la intranet cubana.

Capítulo #2 – Análisis y diseño.

	interactúan con el sistema sin estar autenticados. Tienen básicamente acceso a consultar los materiales públicos que se encuentran en el sistema.
Usuario autenticado	El actor puede acceder a todas las funcionalidades del usuario anónimo pero además puede interactuar con la parte relacionada con el trabajo colaborativo del sistema mediante la emisión de comentarios, votos, documentos y el seguimiento de las actividades que le interesen.
Administrador	El actor puede realizar todas las acciones que corresponden al actor usuario autenticado pero incorpora a su vez funcionalidades relativas al manejo de los documentos y la información del sistema relacionada con este tipo de entidades.
Súper administrador	El actor tiene un nivel de privilegios superior al actor administrador ya que puede realizar las tareas que este realiza pero incorpora a su vez funcionalidades relativas a la administración y la gestión de los permisos de usuarios.

Tabla 2. Justificación de los actores del sistema.

2.4.2. Definición de los casos de uso del sistema

Los casos de uso del sistema (CUS) se utilizan para representar los requisitos de un sistema, describiendo la interacción entre el usuario y el sistema para lograr un objetivo específico. Durante el desarrollo de esta actividad se determinaron un total de 28 casos de uso, que se muestran a continuación:

Capítulo #2 – Análisis y diseño.

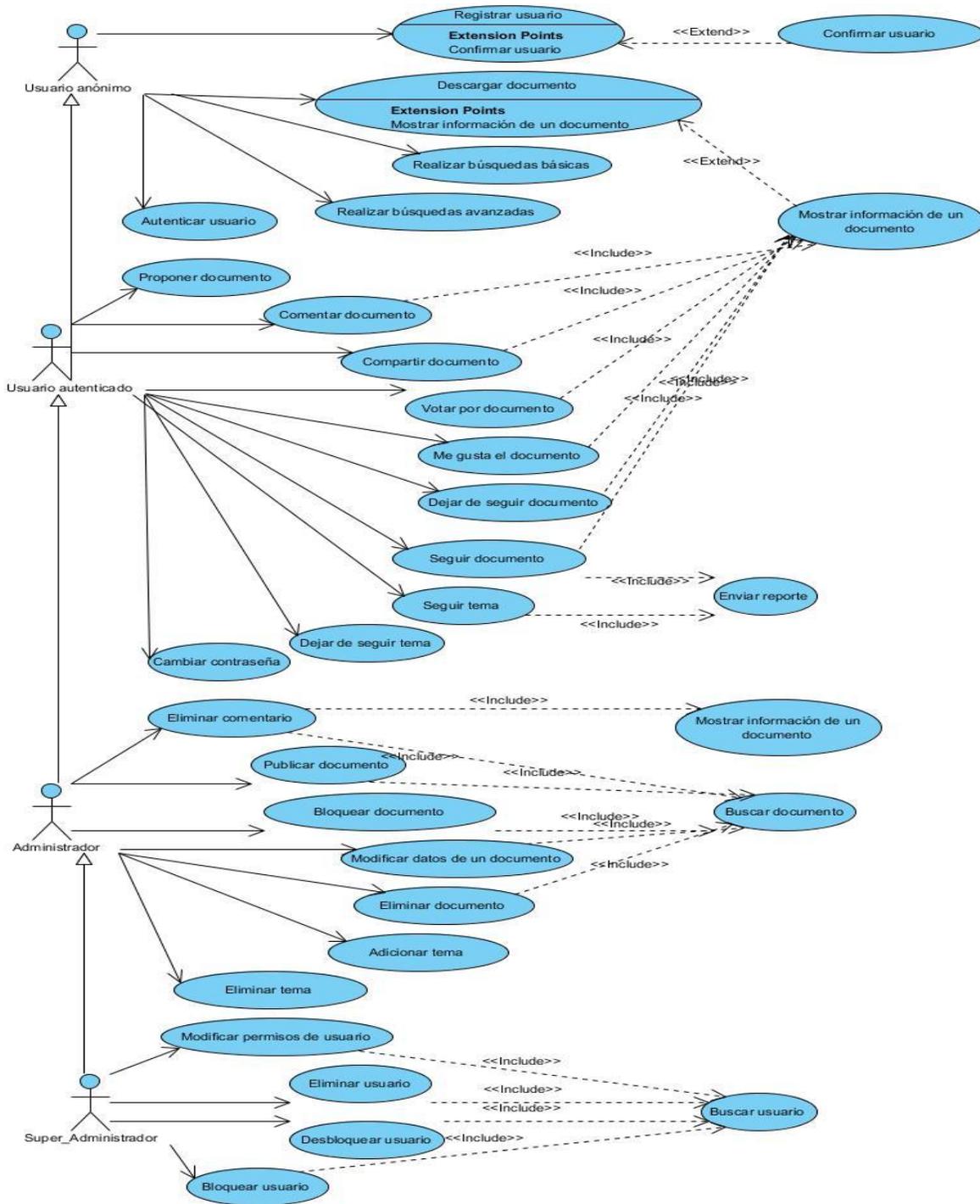


Ilustración 2. Diagrama de casos de uso.

2.4.3. Patrones de casos de uso

La utilización de casos de uso ha evolucionado en un conjunto de patrones que permiten con más precisión reflejar los requisitos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento. Un patrón de caso de uso es un diseño probado en un modelo de casos de uso, junto

Capítulo #2 – Análisis y diseño.

con una descripción del contexto en el cual será usado y las consecuencias que tendrá su aplicación en el modelo [27].

A continuación se enuncian los que fueron utilizados para el modelado del sistema y ejemplos de su utilización:

Concordancia: Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado [28].

✓ **Reuso:** Consta de al menos tres casos de uso. El primero llamado subsecuencia común, modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo. Los otros casos de uso modelan el uso del sistema que comparte la subsecuencia común de acciones. De manera que deben existir al menos dos de ellos.

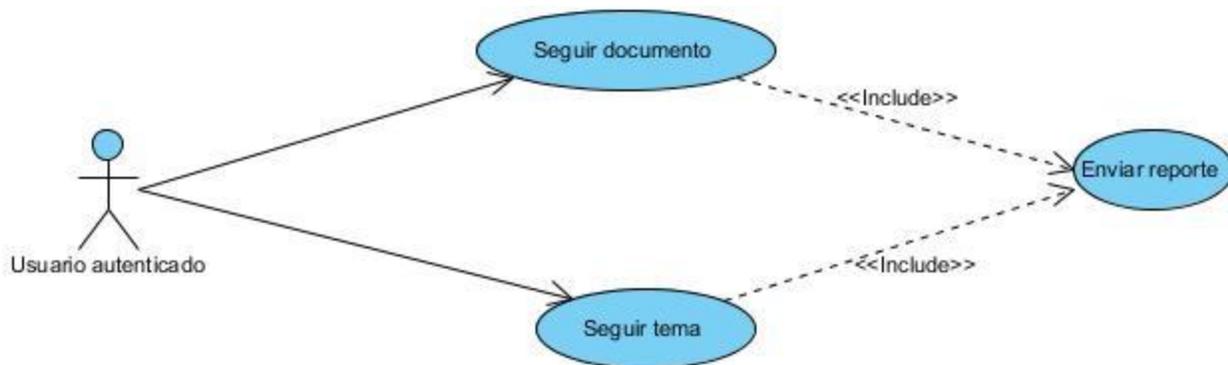


Ilustración 3. Aplicación del patrón reuso.

Extensión concreta o inclusión: Se basa en modelar ambos flujos de trabajo como parte de un caso de uso y como separado, completando el caso de uso por sí solo.

✓ **Extensión:** consiste en la existencia de una relación de extensión entre dos casos de uso. El caso de uso extendido puede ser o no instanciado por el caso de uso base. El caso de uso base puede ser concreto o abstracto. Este patrón se utiliza cuando un flujo puede extender el flujo de otro caso de uso o bien puede ejecutarse dentro de este.

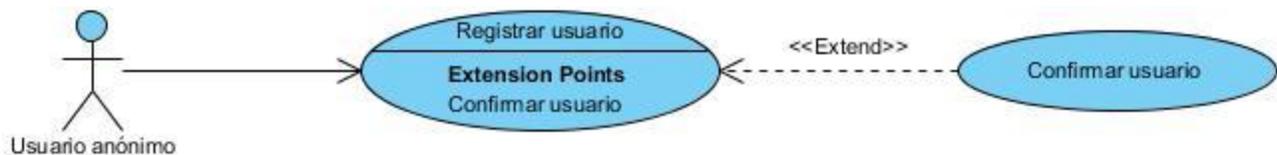


Ilustración 4. Aplicación del patrón extensión.

Capítulo #2 – Análisis y diseño.

2.4.4. Descripción de los casos de uso del sistema

Para una mejor comprensión de cómo fueron descritos los casos de usos a continuación se muestran la descripciones de los casos de usos del sistema **CU5 Proponer documento** y **CU19 Registrar usuario**, para ver las restantes consultar **Anexo 2**.

CU5 Proponer documento.

Objetivo	Proponer que un documento se añada a la base de datos del sistema.	
Actores	Usuario autenticado	
Resumen	El CU se inicia cuando un usuario que se encuentra autenticado en el sistema desea proponer un documento para añadirlo a la colección de documentos de la aplicación e introduce sus datos y el archivo en el sistema, el caso de uso finaliza cuando se ha realizado satisfactoriamente la operación.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción subir documento.	
2.		Muestra una interfaz para realizar esta operación.
3.	Se escriben los principales datos del documento y se introduce el archivo del documento.	
4.		Comprueba que el archivo cumpla con los requisitos necesarios para que sea válido.
5.		Introduce los datos y el archivo del documento en el sistema
6.		Termina el CU.

Capítulo #2 – Análisis y diseño.

Flujos alternos		
4. El archivo proporcionado por el usuario no cumple con las restricciones definidas por el sistema.		
	Actor	Sistema
5.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 3. Descripción CU Proponer documento.

CU19 Registrar usuario

Objetivo	Registrar un usuario al sistema.	
Actores	Usuario anónimo	
Resumen	El CU se inicia cuando un usuario desea registrarse en el sistema y accede a la opción registrarse, el sistema muestra una interfaz donde el usuario introduce sus datos y el sistema lo registra y le envía un correo con un enlace para confirmar su cuenta.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones		
Postcondiciones	El usuario se registra pero no está habilitado	
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción registrarse.	
2.		Muestra una interfaz para realizar

Capítulo #2 – Análisis y diseño.

		esta operación.
3.	Se escriben los datos del usuario.	
4.		Comprueba que los datos cumplan con los requisitos necesarios para que sea válido.
5.		Introduce los datos del usuario en la base de datos.
6.		Envía un correo con un enlace para la confirmación del usuario.
7.		Termina el CU.
Flujos alternos		
4. Los datos proporcionados por el usuario no son correctos.		
	Actor	Sistema
5.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
Relaciones	CU Incluidos	
	CU Extendidos	Confirmar usuario
Requisitos no funcionales		
Asuntos pendientes		

Tabla 4. Descripción CU Registrar usuario.

2.5. Arquitectura del sistema

La arquitectura de *software* puede ser vista como la estructura del sistema en función de la definición de los componentes y sus interacciones. [29]

Los patrones arquitectónicos se pueden definir como la descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. El esquema de solución se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí.

Capítulo #2 – Análisis y diseño.

Para el desarrollo de la aplicación se decidió emplear el patrón arquitectónico Modelo-Vista-Controlador ya que el marco de trabajo de desarrollo Symfony 2 brinda facilidades para el uso de este patrón. Este divide una aplicación interactiva en tres componentes. El modelo contiene la información central y los datos. Las vistas muestran la información y constituyen la interfaz de usuario. Los controladores capturan la entrada del usuario. La ventaja principal de este patrón es que permite dividir la lógica del negocio del diseño.

La aplicación de este patrón se evidencia en cada uno de los bundles que componen la aplicación, ya que en cada uno de ellos en el directorio /views se pueden encontrar todas las vistas, en el /document están cada una de las clases que describen y construyen las entidades y en el directorio /controller se ubican las clases controladoras de cada uno de los bundles conteniendo la lógica del sistema.

2.6. Diagrama de clases del diseño

Los diagramas de clases del diseño, representan las relaciones entre clases que mantienen la información manipulada por el sistema, constituyen además, la interacción de las clases del diseño y sus objetos en la realización de los casos de uso, de forma que de cada uno se desprende un diagrama de clases. A continuación, se muestran los diagramas de clases del diseño de los casos de uso Proponer documento y Registrar usuario en los cuales son empleados los patrones antes definidos seguidos de una breve descripción del mismo.

Capítulo #2 – Análisis y diseño.

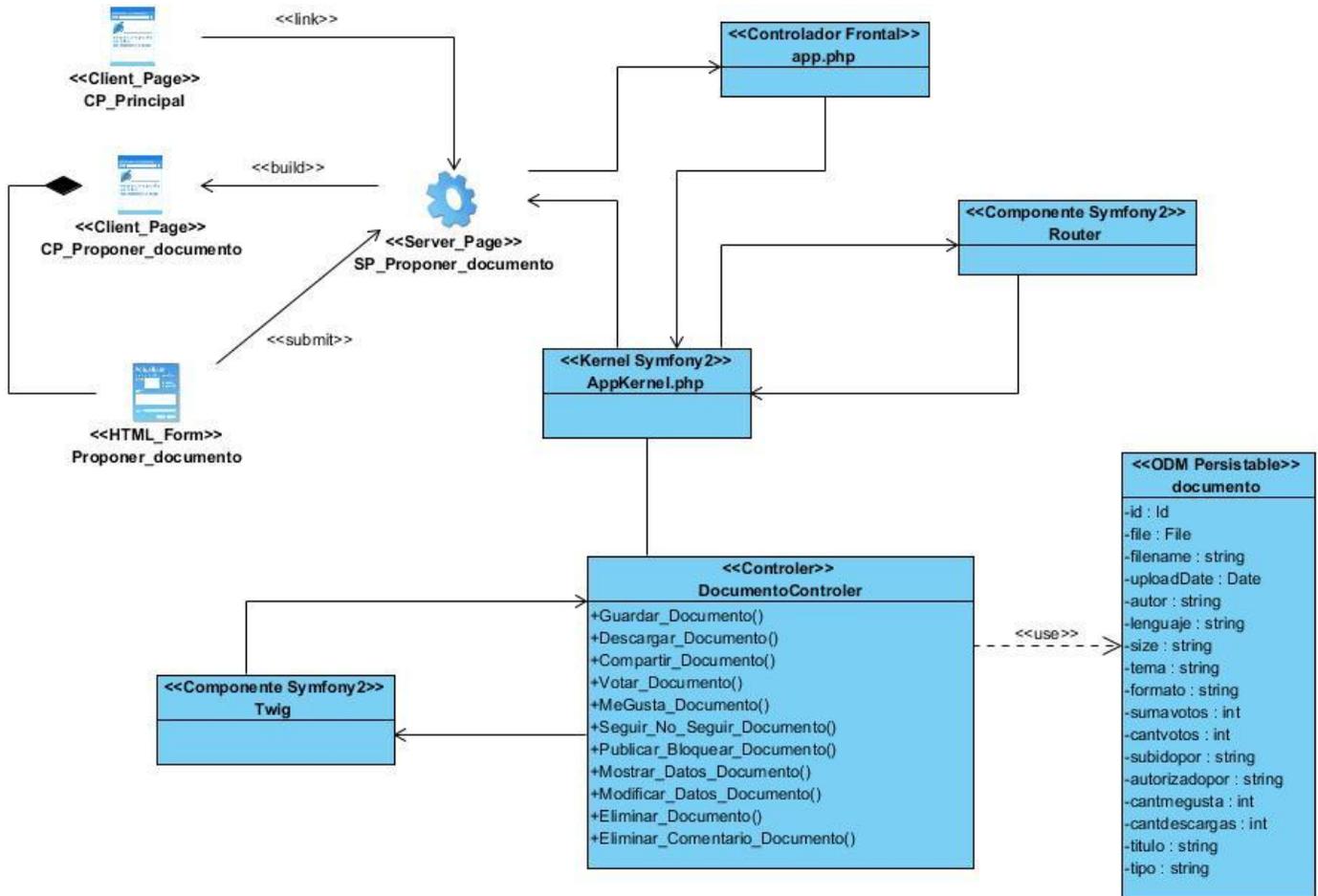


Ilustración 5. Diagrama de clases del diseño CU proponer documento.

En el diagrama anterior, el caso de uso está compuesto por una clase cliente principal (**CP_Principal**) que posee las diferentes opciones relacionadas con los usuarios. Esta clase principal enviará la petición (proponer documento) a la clase servidora **SP_Proponer_documento** que enviará la solicitud al script **app.php**. Este archivo se denomina controlador frontal y es el único punto de entrada a la aplicación. El controlador frontal crea el núcleo de la aplicación mediante una instancia de la clase **AppKernel.php**. A continuación, el controlador frontal envía la información de la petición del usuario al núcleo. Luego el núcleo pregunta al **Router** que es el componente de enrutamiento, qué controlador se debe ejecutar para atender la petición del usuario. Una vez localizado el controlador (**DocumentoControler**) se realiza la petición proponer documento. El controlador haciendo uso del motor de plantillas Twig envía el nombre de una plantilla al núcleo. Este envía la plantilla a la clase servidora **SP_Proponer_documento** que construye la página cliente **CP_Proponer_documento**, devolviendo al usuario la página creada a partir de la plantilla. La página cliente **CP_Proponer_documento** posee un formulario (**Proponer_documento**) que permite el ingreso de los datos por parte del usuario. Los datos de este formulario son enviados a la página servidora **SP_Proponer_documento**, quien a su vez enviará la información al controlador frontal, donde siguiendo el mismo proceso descrito anteriormente llegan los datos al controlador

Capítulo #2 – Análisis y diseño.

DocumentoControler, quien validará estos datos y con el método **Guardar_Documento()** persistirá los datos en la base de datos y haciendo uso de la clase **documento**. Nuevamente el controlador haciendo uso del motor de plantillas **Twig** envía el nombre de una plantilla al núcleo. Este enviará la plantilla a la clase servidora **SP_Proponer_documento** que construye la página cliente **CP_Proponer_documento**, devolviendo al usuario la página creada a partir de la plantilla y donde muestra un mensaje dependiendo de la validez de los datos proporcionados.

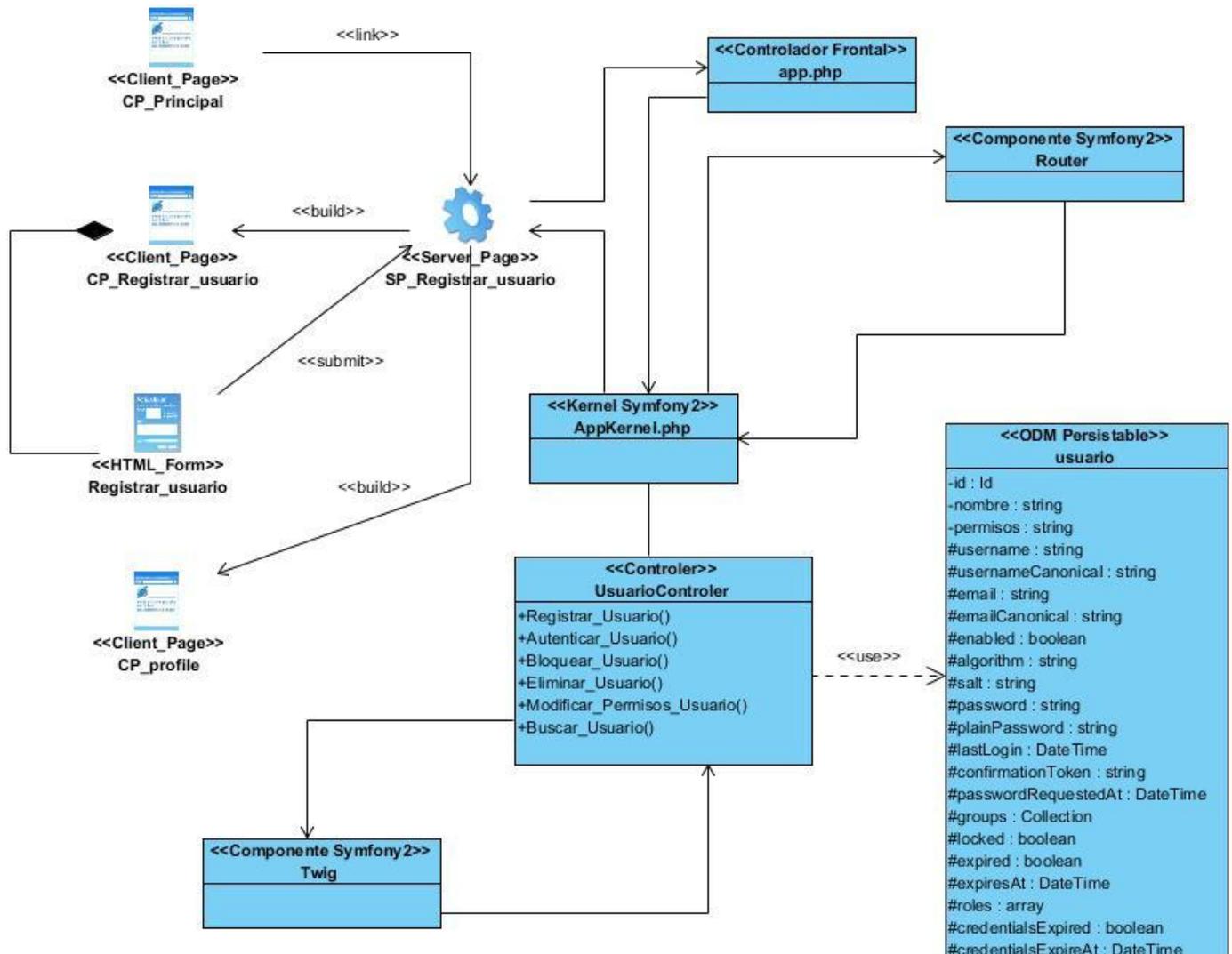


Ilustración 6. Diagrama de clases del diseño CU registrar usuario.

En el diagrama anterior, el caso de uso está compuesto por una clase cliente principal (**CP_Principal**) que posee las diferentes opciones relacionadas con los usuarios. Esta clase principal enviará la petición (registrar usuario) a la clase servidora **SP_Registrar_usuario**, y esta a su vez enviará la solicitud al script **app.php**. Este archivo se denomina controlador frontal y es el único punto de entrada a la aplicación. El controlador frontal crea el núcleo de la aplicación mediante una instancia de la clase **AppKernel.php**. A continuación, el controlador frontal envía la información de la petición del usuario al núcleo. Luego el

Capítulo #2 – Análisis y diseño.

núcleo pregunta al **Router** que es el componente de enrutamiento, qué controlador se debe ejecutar para atender la petición del usuario. Una vez localizado el controlador (**UsuarioControler**) se realiza la petición registrar usuario. El controlador haciendo uso del motor de plantillas **Twig** envía el nombre de una plantilla al núcleo. Este envía la plantilla a la clase servidora **SP_Registrar_usuario** que construye la página cliente **CP_Registrar_usuario**, devolviendo al usuario la página creada a partir de la plantilla. La página cliente **CP_Registrar_usuario** posee un formulario (**Registrar_usuario**) que permite el ingreso de los datos por parte del usuario. Los datos de este formulario son enviados a la página servidora **SP_Registrar_usuario**, quien a su vez enviará la información al controlador frontal, donde siguiendo el mismo proceso descrito anteriormente llegan los datos al controlador **UsuarioControler**, quien validará estos datos y con el método **Registrar_Usuario()** persistirá los datos en la base de datos y haciendo uso de la clase **usuario**. Nuevamente el controlador haciendo uso del motor de plantillas **Twig** envía el nombre de una plantilla al núcleo. Este enviará la plantilla a la clase servidora **SP_Registrar_usuario** que construye la página cliente **CP_Registrar_usuario** o **CP_profile**, dependiendo de la validación de los datos proporcionados, ya sea la **CP_profile** en caso de ser los datos validos o **CP_Registrar_usuario** en caso contrario. De modo general devuelve al usuario la página creada a partir de la plantilla.

Para ver los restantes diagramas de clases del diseño, consultar **Anexo 3**.

2.6.1. Patrones de diseño empleados durante el desarrollo del sistema

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de *software*, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular.

En palabras más simples un patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas. Son menores en escala que los patrones arquitectónicos, y tienden a ser independientes de los lenguajes y paradigmas de programación.

Durante el desarrollo del sistema se usaron los patrones *GRASP (General Responsibility Assignment Software Patterns)* los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

A continuación se muestra una selección de estos patrones los cuales se utilizaron durante la realización del diseño de la aplicación:

Experto: es uno de los patrones más usado en la asignación de responsabilidades; es un principio básico utilizado en el diseño orientado a objetos. Expresa que siempre se debe asignar una responsabilidad al experto en información, es decir a la clase que cuenta con la información necesaria para llevar a cabo la funcionalidad.

Capítulo #2 – Análisis y diseño.

La clase Documento_Controller funciona como experta en información para llevar a cabo el trabajo con los documentos del sistema como por ejemplo en el caso de uso proponer documento esta clase obtiene toda la información necesaria para archivar relacionada con el documento.

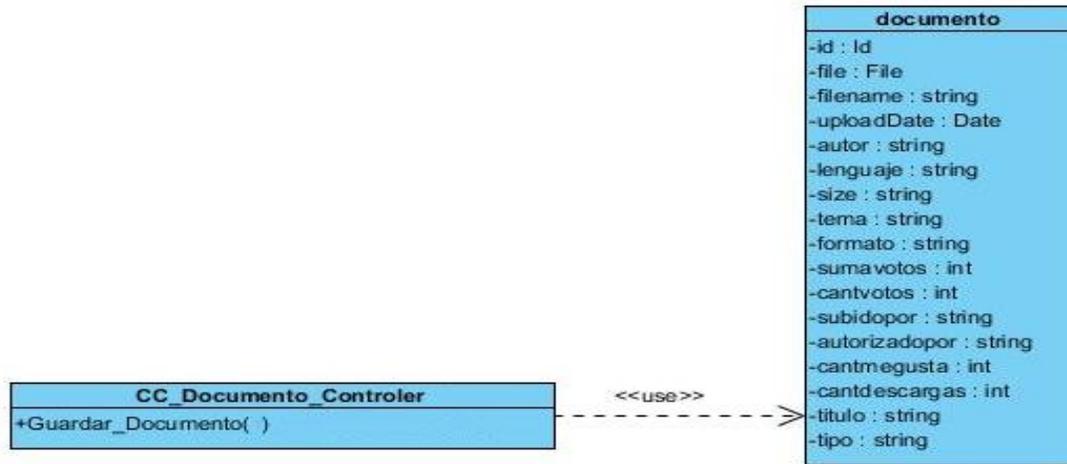


Ilustración 7. Aplicación del patrón experto.

Creador: este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. La intención básica del mismo es encontrar un creador que necesite conectarse al objeto creado en alguna situación, eligiéndolo como el creador.

Este patrón es usado principalmente para el trabajo con las entidades que se guardan en la base de datos. Ejemplo de su uso es en el caso de uso proponer documento ya que la clase Documento_Controller es la encargada de crear los objetos de tipo documento que luego serán almacenadas en la base de datos debido a que obtiene todos los datos necesarios para la inicialización de los mismos.

Capítulo #2 – Análisis y diseño.

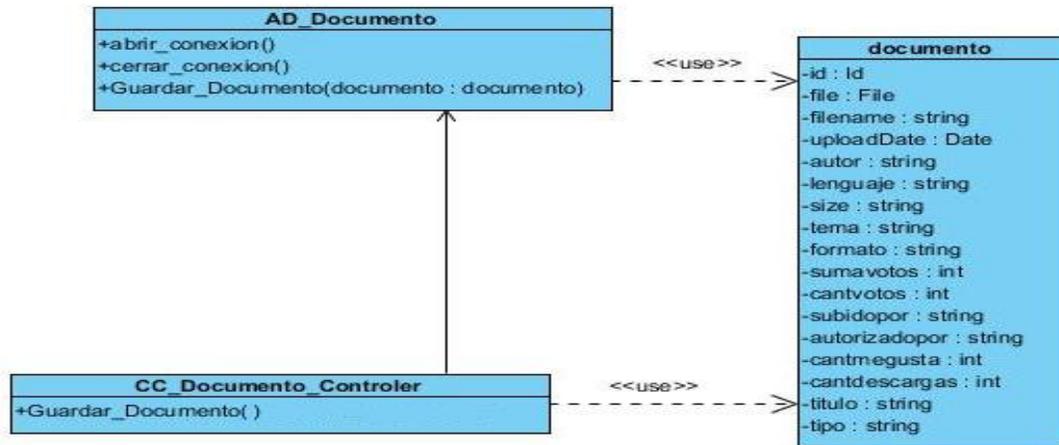


Ilustración 8. Aplicación del patrón creador.

Bajo Acoplamiento: el patrón bajo acoplamiento impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve a los resultados negativos que puede producir un acoplamiento alto. No soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio. El mismo no se puede considerar de manera aislada a otros patrones como el Experto o el de Alta Cohesión, sino que necesita incluirse como uno de los diferentes principios de diseño que influyen en una elección, al asignar una responsabilidad.

Se evidencia en todo el sistema ya que las clases controladoras del sistema no se relacionan entre sí, lo que disminuye las dependencias entre las mismas.

Alta Cohesión: este patrón es un principio a tener en mente durante todas las decisiones de diseño. Constituye un objetivo subyacente a tener en cuenta continuamente. Es un principio evaluativo que aplica un diseñador mientras evalúa todas las decisiones de diseño. Como beneficios que este aporta se pueden mencionar:

- 1 Se incrementa la claridad y facilita la comprensión del diseño.
- 2 Se simplifican el mantenimiento y las mejoras.
- 3 Se soporta a menudo bajo acoplamiento.

Se puede observar claramente en el sistema ya que cada clase controladora se ajusta a manejar solo las responsabilidades correspondientes a las entidades con las que se relaciona, además para cada página cliente existe una página servidora encargada de manejar sus solicitudes poniéndose de manifiesto el patrón de diseño alta cohesión.

Controlador: es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. La mayor parte de los sistemas reciben eventos de entrada externa, en cualquiera

Capítulo #2 – Análisis y diseño.

de los casos que puedan existir, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada.

Se pone de manifiesto en todo el sistema debido a que cada uno de los eventos generados por el usuario es redirigido a una clase controladora que realice las operaciones solicitadas, pero siempre manteniendo las clases controladoras sin saturarse es decir siempre manteniendo la alta cohesión.

2.7. Diagramas de interacción del diseño

Los diagramas de secuencia, describen las interacciones entre un grupo de objetos mostrando de forma secuencial los envíos de mensajes entre estos. A continuación, se presenta el diagrama de secuencia correspondiente a los casos de uso Proponer documento y Registrar usuario.

El diagrama que se muestra a continuación pertenece al caso de uso **proponer documento**. Es te comienza con la selección de la opción proponer documento por parte del usuario desde la página principal **CP_Principal**. Esta petición es enviada a la clase controladora **CC_Documento_Controler**, la cual construye y muestra la página **CP_Proponer_documento** que permitirá que el usuario ingrese el documento en formato digital y sus datos asociados. Luego el usuario ingresa los datos y envía la información, la información se valida y si los datos son correctos la página **CP_Proponer_documento** envía los datos a la clase controladora **CC_Documento_Controler**, en caso contrario muestra un mensaje donde informa que los datos ingresados son incorrectos. Luego de enviados los datos a la clase controladora **CC_Documento_Controler** esta comprueba en la base de datos si ya existe ese documento, si no es así se guardan los datos, en caso contrario la clase **CC_Documento_Controler** envía a la página **CP_Proponer_documento** un mensaje informando del error.

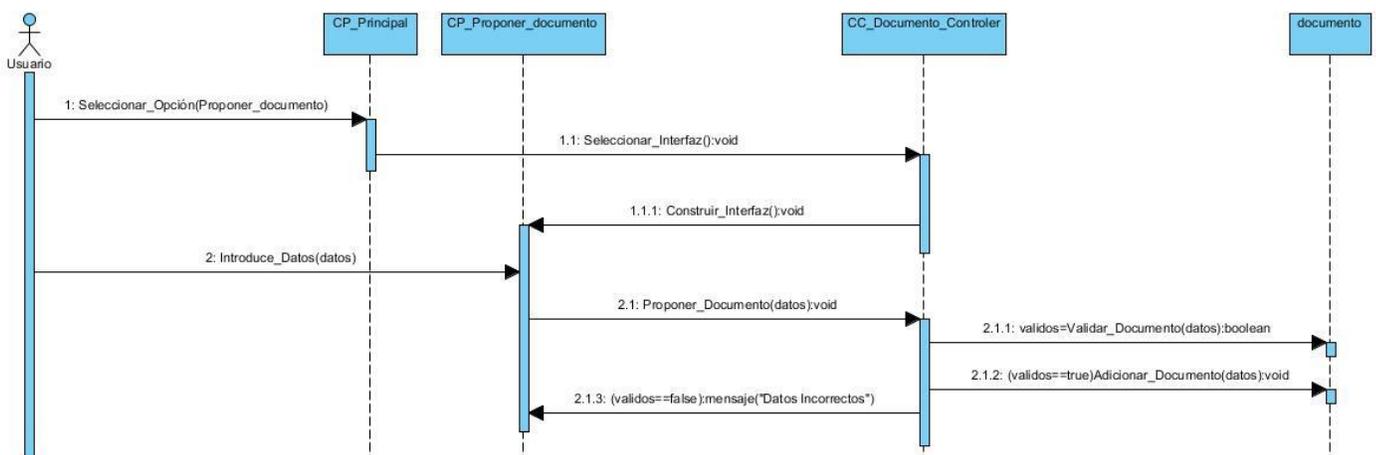


Ilustración 9. Diagrama de secuencia CU proponer documento.

El diagrama que se muestra a continuación pertenece al caso de uso **registrar usuario**. Es te comienza con la selección de la opción registrarse por parte del usuario desde la página principal **CP_Principal**. Esta petición es enviada a la clase controladora **CC_Usuario_Controler**, quien construye y muestra la

Capítulo #2 – Análisis y diseño.

página **CP_Registrar_usuario** que permitirá que el usuario ingrese sus datos. Luego el usuario ingresa los datos y envía la información, la información se valida y si los datos son correctos la página **CP_Registrar_usuario** envía los datos a la clase controladora **CC_Usuario_Controller**, en caso contrario muestra un mensaje donde informa que los datos ingresados son incorrectos. Luego de enviados los datos a la clase controladora **CC_Usuario_Controller** esta comprueba en la base de datos si ya existe este usuario, si no es así se guardan los datos, en caso contrario la clase **CC_Usuario_Controller** envía a la página **CP_Registrar_usuario** un mensaje informando del error.

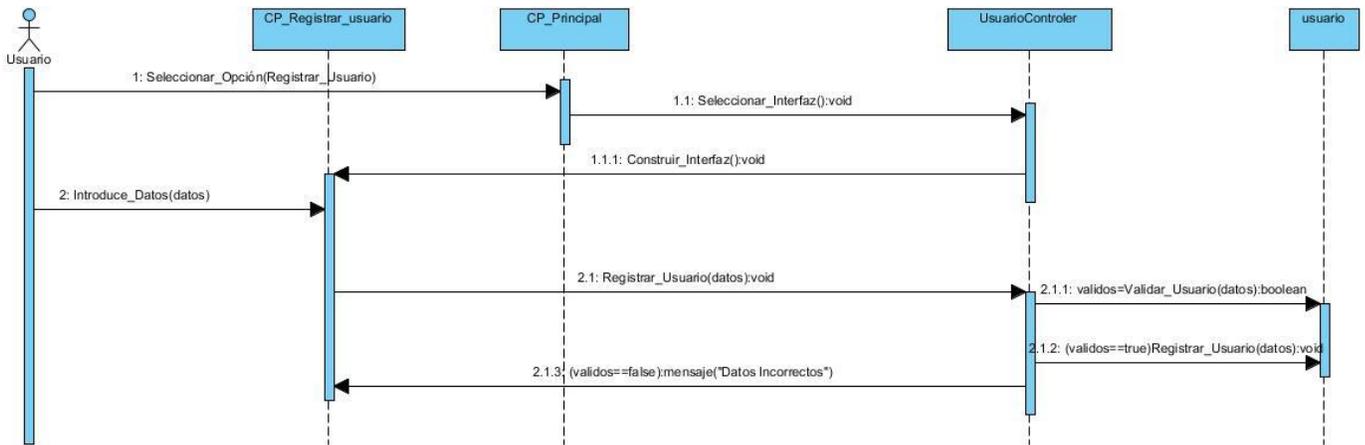


Ilustración 10. Diagrama de secuencia CU registrar usuario.

Para ver los restantes diagramas de secuencia consultar **Anexo 4**.

2.8. Diseño de la base de datos

Las bases de datos desempeñan un papel crucial en casi todas las áreas de desarrollo de *software*. Su diseño es una de las tareas más importantes en la construcción de un sistema que requiera manejar la persistencia de la información. El sistema que se propone hace uso de una base de datos, ya que es necesario persistir una cantidad considerable de información relacionada con los documentos y los usuarios principalmente.

MongoDB es un gestor de base de datos no relacional que se basa en colecciones. Cuando se dice que se *basa en colecciones*, significa que los datos se agrupan en conjuntos llamados "colecciones". Cada colección tiene un nombre único en la base de datos y puede contener un número ilimitado de documentos. Una colección es análoga a una tabla, excepto que no tienen un esquema definido.

La colección de datos **usuario** almacenará los datos correspondientes a los usuarios registrados en el sistema.

Capítulo #2 – Análisis y diseño.

La colección **documento** almacenará los datos asociados a cada documento almacenado en la aplicación. Entiéndase por documento: un manual, libro, tutorial, publicación, revista, tesis, presentación o cualquier otro archivo en formato digital que brinde información útil para los usuarios.

La colección **tema** almacenará cada uno de los temas en los que se pueden categorizar los documentos que se encuentran almacenados en la aplicación.

La colección **seguirTema** almacenará la relación de seguimiento que se establece entre los usuarios que se hagan seguidor de un tema determinado y el tema.

La colección **voto** almacenará los votos efectuados por los usuarios sobre un documento determinado.

La colección **comentario** almacenará los comentarios efectuados por los usuarios acerca de un documento determinado.

La colección **meGusta** registrará los documentos que le gustan a cada usuario.

La colección **seguirDocumento** almacenará la relación de seguimiento que se establece entre los usuarios que se hagan seguidor de un documento determinado y el documento.



Ilustración 11. Modelo físico de datos para MongoDB.

Capítulo #2 – Análisis y diseño.

2.9. Diagrama de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del *software* se trazan en esos nodos.



Ilustración 12. Diagrama de despliegue.

La aplicación será desplegada de la siguiente manera: la PC_Cliente a través del navegador utilizará el protocolo HTTP para la conexión con el servidor web que tendrá alojada la aplicación web Plataforma Colaborativa. Esta aplicación es la encargada de realizar la recuperación y publicación de la información en la web, que a través del protocolo de comunicación Mongo Wire Protocol (MONGO) garantizará la persistencia de los datos en el servidor de base de datos con el gestor MongoDB ejecutándose.

2.10. Conclusiones

Se definieron importantes aspectos correspondientes al desarrollo de la solución propuesta lo que permitió comprender mejor el funcionamiento y flujo de información en el sistema, teniéndose en cuenta el desarrollo de una solución que cumpla con las expectativas del cliente. Se identificaron los procesos y las personas que van a interactuar con la aplicación, las funcionalidades y los requisitos no funcionales del *software*, sentando las bases para las siguientes etapas del desarrollo de la aplicación.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DE LA PLATAFORMA COLABORATIVA PARA LA PUBLICACIÓN DE DOCUMENTOS DIGITALES

A partir de los resultados obtenidos durante el diseño de la aplicación, se definen los estándares de codificación empleados, además de los componentes y subsistemas de implementación lo que permite comprender como se organizan de acuerdo con los mecanismos de estructuración utilizados en el entorno de implementación y cómo dependen unos de otros. Además se documentan las pruebas realizadas para la validación del software como elemento crítico en la garantía de la calidad y para la revisión final del cumplimiento de las especificaciones del diseño y de la codificación.

3.1. Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiese escrito todo el código de una sola vez. La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de *software*. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutinas.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del *software* y para obtener un buen rendimiento. Además, si se aplica de forma continuada un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas, y posteriormente se efectúan revisiones del código de rutinas, caben muchas posibilidades de que un proyecto de *software* se convierta en un sistema de *software* fácil de comprender y de mantener. [30]

Para la implementación de la plataforma colaborativa para la publicación de documentos digitales se estandarizó el código, lo que permite que exista legibilidad y organización para su mantenimiento. A continuación se especifican los estándares de código a utilizar en la construcción del sistema, los cuales se basan en las normas definidas por los estándares PSR-0, PSR-1 y PSR-2, utilizados para la codificación de aplicaciones que utilizan el marco de trabajo de desarrollo Symfony2.

Capítulo #3 – Implementación y pruebas.

El nombre de las clases debe ser declarado utilizando el estilo **StudyCaps** que permite hacer uso de mayúsculas y minúsculas indiscriminadamente.

```
1 ....class EjEmpIO()
2 ....{
3 ....//Bloque de instrucciones
4 ....}
```

El nombre de los procedimientos debe ser declarado utilizando el estilo **camelCase**, este define que el nombre de las funciones se escriba con la primera palabra en minúscula, y las restantes que comiencen con mayúscula.

```
1 ....function miMetodo()
2 ....{
3 ....//Bloque de instrucciones
4 ....}
```

Todo el código PHP debe estar encerrado en las etiquetas `<?php ?>` pero en caso de que el fichero solo contenga código escrito en PHP se omitirá la etiqueta de cierre `?>`

1....//Código no PHP	1....<?php
2....<?php	2....//Código PHP
3....//Código PHP	
4....?>	

El código debe tener una indentación de cuatro espacios sin tabulaciones y la longitud máxima que puede alcanzar una línea es de ochenta caracteres, las que excedan este tamaño deben ser divididas en varias líneas con un tamaño que no viole esta restricción. Además no debe escribir más de una instrucción por línea y no deben existir espacios en blanco al final de las líneas que no estén vacías.

```
1....//Código(longitud máxima 80 caracteres)
```

Las constantes de PHP **true**, **false** y **null** deben estar escritas en minúsculas.

```
1....$variable1=true;
```

Cuando en el fichero se define el espacio de nombre debe existir una línea en blanco a continuación de la declaración. Además cuando están presentes todas las declaraciones de uso se escriben a

Capítulo #3 – Implementación y pruebas.

continuación de la definición de espacio de nombre, se debe utilizar la palabra reservada **use** una vez por cada declaración y debe concluir con una línea en blanco.

```
1....namespace Camino\Paquete;  
2....//Línea en blanco  
3....use MiClase;  
4....use OtraClase;  
5....//Línea en blanco
```

Se debe declarar la visibilidad de todos los atributos de las clases y no debe de existir más de un atributo por línea de código.

```
1....class MiClase  
2....{  
3....public $atributo1;  
4....public $atributo2;  
5....}
```

Se debe declarar la visibilidad de todos los procedimientos y no deben existir espacios en blanco a continuación de los nombre de los mismos, ni a continuación del paréntesis abierto, ni antes del paréntesis cerrado de los parámetros. Las llaves deben colocarse una en cada línea de forma que la que abre se encuentre a continuación de la declaración del método y la que cierra a continuación del bloque de instrucciones.

```
1....public function miFuncion($arg1, $arg2, $arg3)  
2....{  
3....//Bloque de instrucciones  
4....}
```

Para las estructuras de control se define que debe existir un espacio en blanco a continuación de la palabra reservada que define la estructura de control, no debe existir un espacio en blanco ni después del paréntesis abierto ni antes del paréntesis cerrado y debe existir un espacio en blanco entre el paréntesis que cierra y la llave que abre el bloque de instrucciones. El bloque de instrucciones debe estar indentado un nivel a la derecha y a continuación debe aparecer la llave que cierra.

Capítulo #3 – Implementación y pruebas.

```
1....if ($condicion1) {
2....//Bloque de instrucciones
3....} elseif ($condicion2) {
4....//Bloque de instrucciones
5....} else {
6....//Bloque de instrucciones
7....}

1....while ($condicion) {
2....//Bloque de instrucciones
3....}

1....for ($i = 0; $i < 10; $i++) {
2....//Bloque de instrucciones
3....}
```

Se debe añadir los bloques de documentación al principio de cada clase y procedimiento de la forma que se muestra a continuación.

```
/**
 * Descripción de la clase.
 */
1....class MiCLASE
2....{
3....//Código de la clase
/**
 * @param string $parametro descripción del parámetro
 *
 * @return ValorQueDevuelve
 */
4....public function miFuncion($parametro)
5....{
6....//Bloque de instrucciones
7....}
```

3.2. Diagrama de componentes

Capítulo #3 – Implementación y pruebas.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Los componentes representan los elementos del *software* resultantes del proceso de implementación.

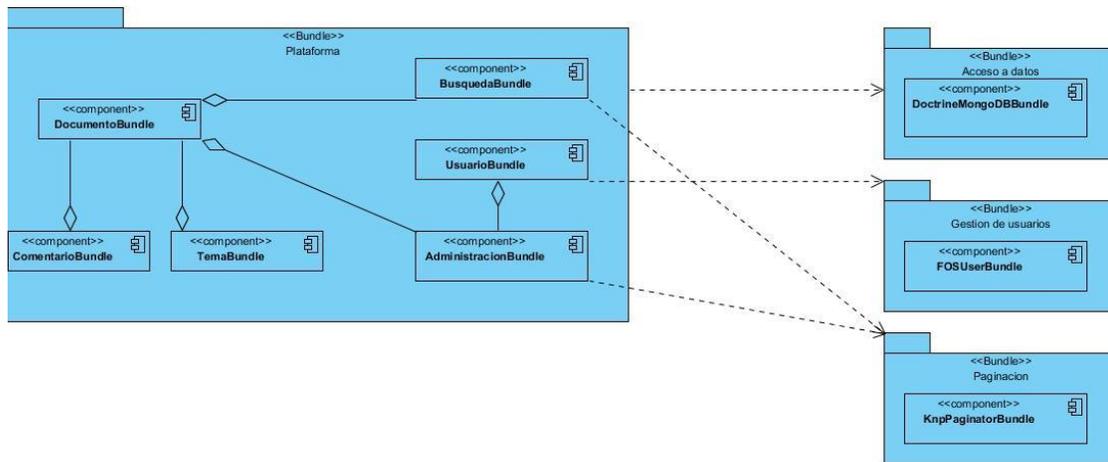


Ilustración 13. Diagrama de componentes.

Como se puede observar en el diagrama anterior el sistema desarrollado está empaquetado en un bundle principal que está integrado a su vez por bundles internos relacionados entre sí (ver figuras de la 13 a la 19). También se representan las dependencias del sistema las cuales son: DoctrineMongoDBBundle utilizado para el acceso a datos y el trabajo con el gestor de base de datos MongoDB, KnpPaginatorBundle utilizado principalmente para la paginación de resultados y el FOSUserBundle empleado principalmente en la gestión de usuarios. Cada bundle interno está compuesto generalmente por una clase controladora, los documentos de MongoDB encargados del mapeo de los datos y las interfaces.

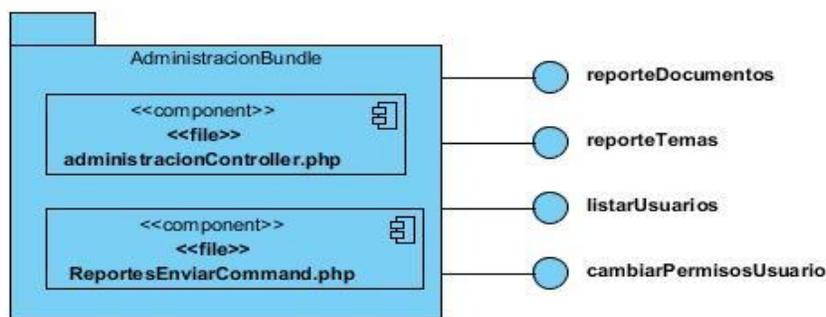


Ilustración 14. Diagrama de componentes del bundle de administración.

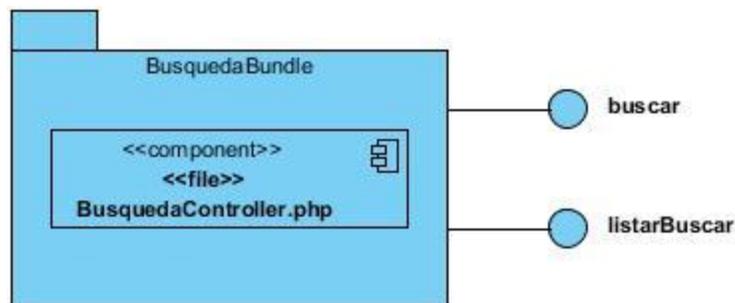


Ilustración 15. Diagrama de componentes del bundle de búsqueda.

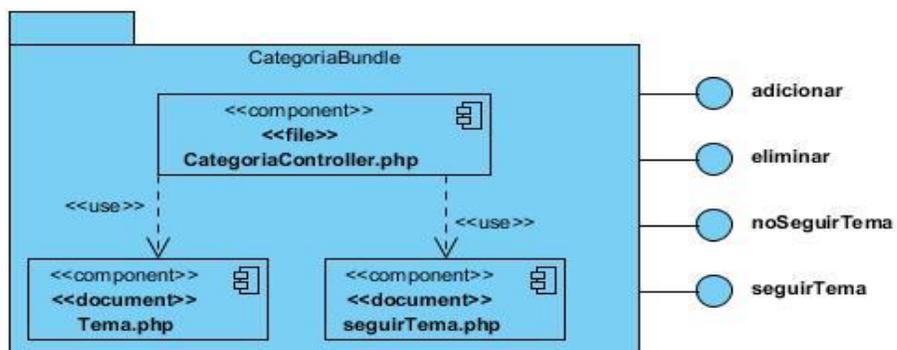


Ilustración 16. Diagrama de componentes del bundle de temas.

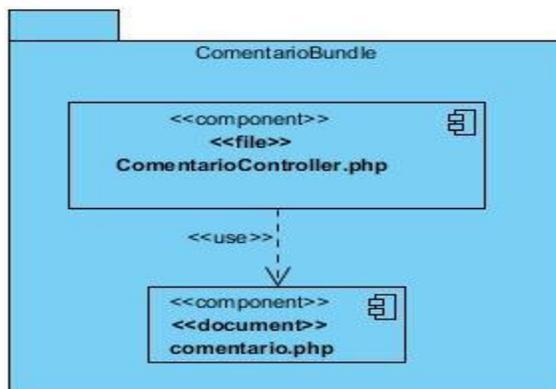


Ilustración 17. Diagrama de componentes del bundle de comentarios.

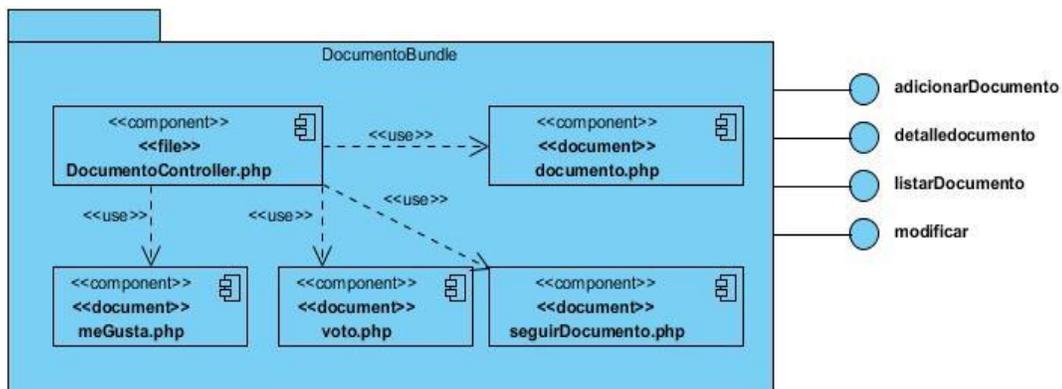


Ilustración 18. Diagrama de componentes del bundle de documentos.

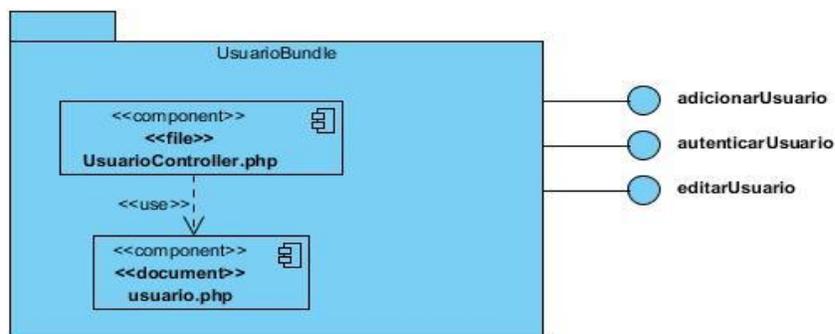


Ilustración 19. Diagrama de componentes del bundle de usuario.

3.3. Interfaces principales de la aplicación

Con el objetivo de obtener una mejor comprensión visual de la aplicación a continuación se presentan algunas imágenes de las interfaces de la aplicación.

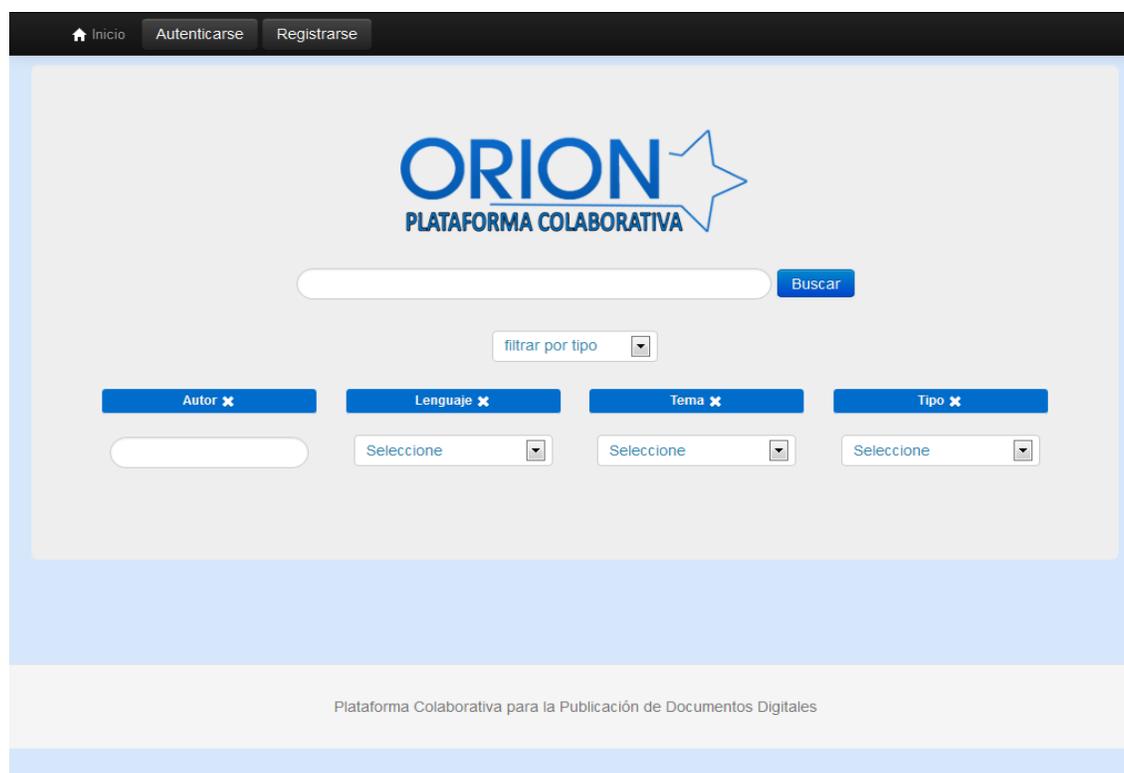


Ilustración 20. Interfaz para la búsqueda de documentos con filtros.

La interfaz anterior permite a los usuarios que interactúan con el sistema realizar búsquedas básicas y avanzadas de documentos introduciendo un criterio de búsqueda y además si desean pueden añadir algunos filtros que posibilitarán la obtención de un resultado más preciso.

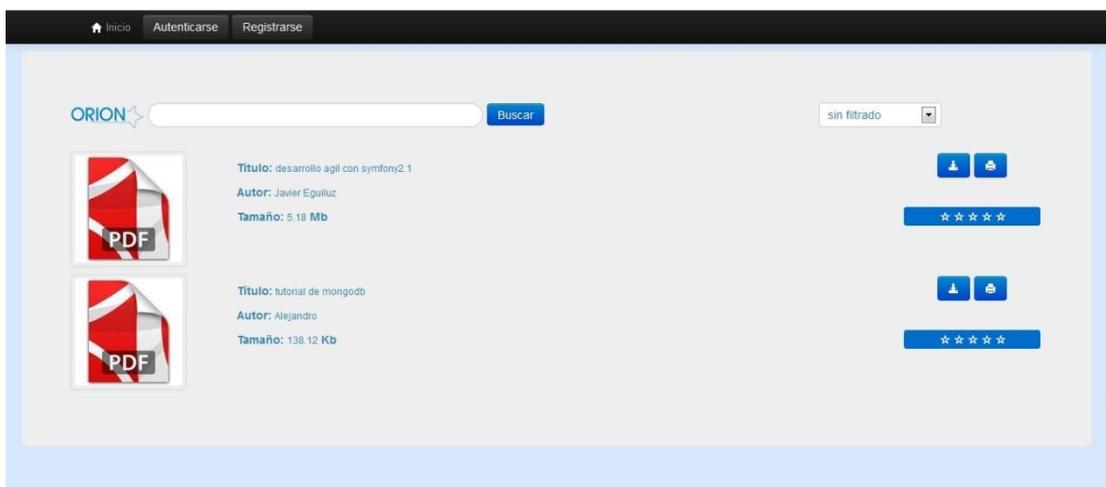


Ilustración 21. Resultados de búsquedas paginado.

La interfaz anterior es la encargada de mostrar a los usuarios que interactúan con el sistema los resultados de las búsquedas realizadas y permite descargar estos documentos. Además si así lo desean dando clic en un documento pueden ver los datos asociados al mismo y si el usuario se encuentra autenticado en el sistema podrá además realizar algunas acciones sobre estos documentos.

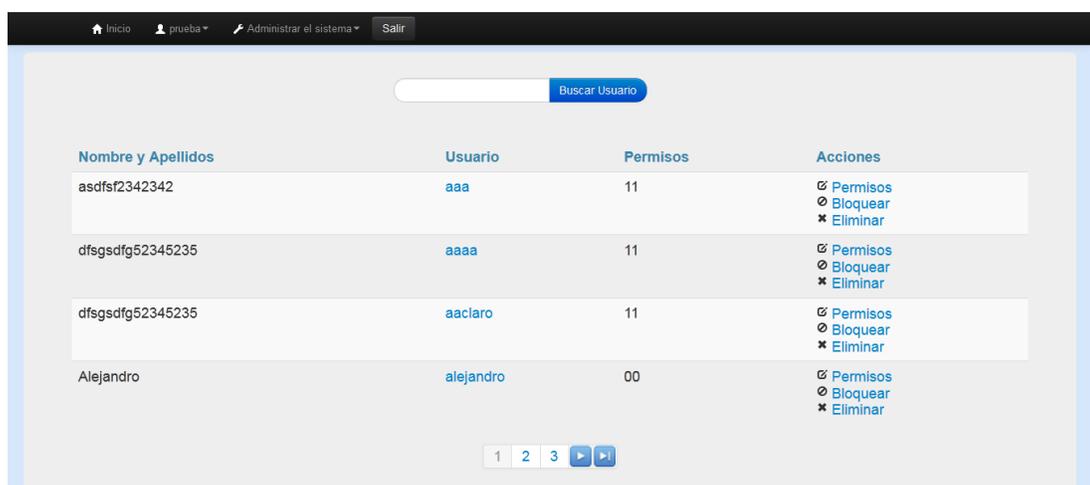


Ilustración 22. Interfaz para la gestión de usuarios y permisos.

La interfaz anterior permite a los administradores del sistema gestionar a los usuarios registrados en el mismo, de manera que brinda la opción de buscar un usuario y realizar acciones sobre estos como son: editar los permisos, eliminar y bloquear a los usuarios.

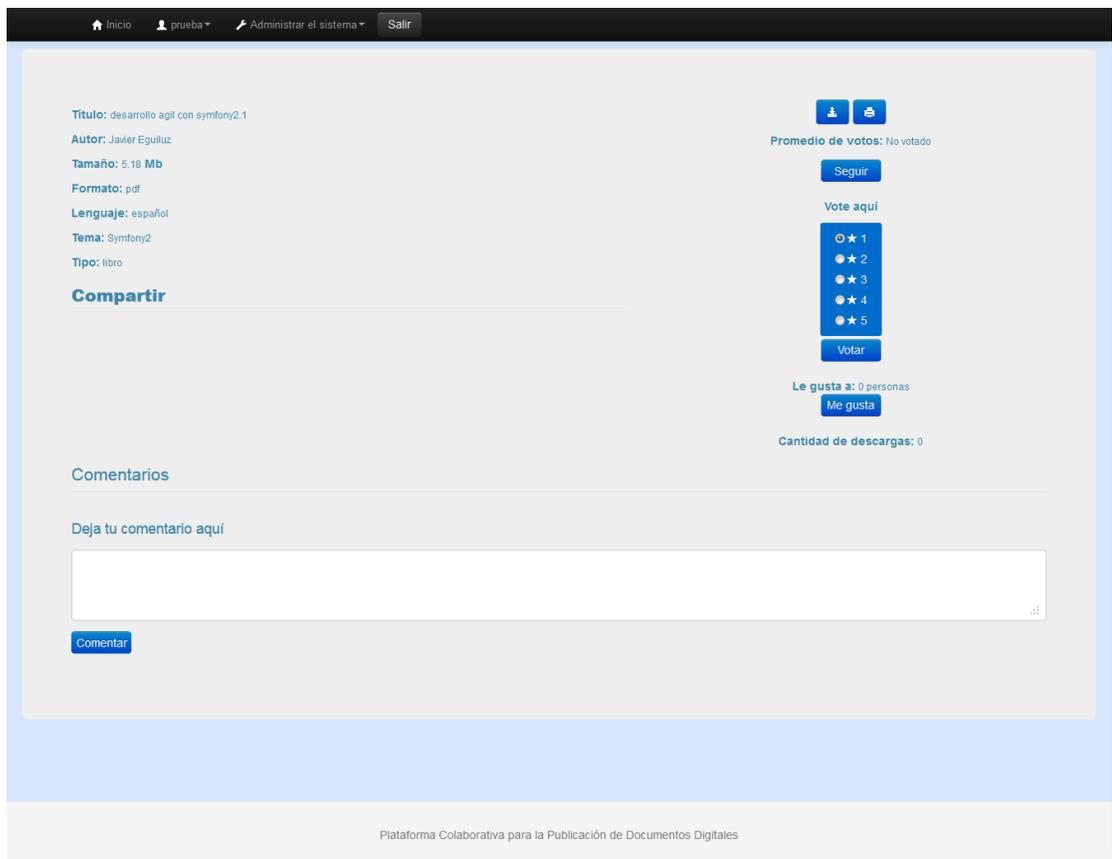


Ilustración 23. Interfaz que muestra la información de un documento.

La interfaz anterior es la encargada de mostrar a los usuarios que interactúan con el sistema los datos asociados a un documento y además permitirá realizar algunas acciones sobre estos documentos como son: descargar, votar, opinar si le gusta, realizar algún comentario y hacerse seguidor del documento, de manera tal que se le enviará a su correo reportes de actividades relacionadas con el documento en el sistema.

3.4. Pruebas de software

Desarrollar determinado *software* debe ir acompañado de una actividad que garantice su calidad, para asegurarse de ello la prueba de *software* es una etapa vital, pues representa una revisión final de las especificaciones del diseño y de la codificación, entre sus principales objetivos se encuentran:

Verificar la interacción entre los elementos del sistema.

Verificar la integración apropiada de los componentes.

Validar que se satisfacen los requerimientos.

Identificar los defectos y corregirlos antes de dar por terminado el *software*.

Capítulo #3 – Implementación y pruebas.

Los casos de prueba intentan demostrar que el *software* es funcionalmente operativo, que las entradas son aceptadas y que se produce un resultado correcto, además de asegurar que se mantiene la integridad de la información.

3.4.1. Pruebas de funcionalidad

Para la validación de un *software* de manera general se utilizan principalmente dos métodos de prueba: **caja blanca** y **caja negra**. La pruebas de caja blanca basa su funcionamiento en evaluar la información sobre cómo el *software* ha sido diseñado y codificado. El método caja negra actúa sobre la validación de los requisitos funcionales y se aplican a la interfaz del *software* para examinar algún aspecto funcional de un sistema, por lo que también se les denominan pruebas funcionales. Las pruebas de caja negra intentan encontrar errores de las siguientes categorías [31]:

- ✓ Funciones incorrecta o ausente.
- ✓ Errores de interfaz.
- ✓ Errores en estructuras de datos o en accesos a bases de datos externos.
- ✓ Errores de rendimiento y errores de inicialización y de terminación

También el método caja negra posee dos criterios fundamentales: **particiones de equivalencia** y **análisis de valores límite**.

Partición equivalente: Se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Este método intenta dividir el dominio de entrada de un programa en un número. El diseño de casos de prueba según esta técnica consta de dos pasos [31]:

1. Identificar las clases de equivalencia: Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada de un programa.
2. Identificar los casos de prueba. finito de clases de equivalencia: El objetivo es minimizar el número de casos de prueba, así cada caso de prueba debe considerar tantas condiciones de entrada como sea posible.

Esta técnica se basa en recoger todas las combinaciones de escenarios posibles a ejecutar, contienen la descripción de todas las variables o datos de entrada al sistema. Son un conjunto de condiciones mediante las cuales se comprueba si los requisitos funcionan correctamente [31].

Se decide utilizar en la realización de las pruebas funcionales a la plataforma colaborativa para la publicación de documentos digitales el método de caja negra con la aplicación del criterio partición

Capítulo #3 – Implementación y pruebas.

equivalente. Este tipo de pruebas se aplicaron con el objetivo de localizar fallas funcionales en el sistema, al identificar situaciones en las que las respuestas de esta a determinadas acciones del usuario no se apegan a las especificaciones establecidas. Se ejecutaron pruebas a todos los casos de uso con diferentes entradas del usuario para determinar que los resultados obtenidos fueran consistentes bajo cualquier situación lo que permitió comprobar el cumplimiento de cada uno de ellos y el cumplimiento de los requisitos establecidos.

A continuación se detallan los casos de prueba realizadas a los casos de uso **Proponer Documento** y **Registrar Usuario**. Para ver los restantes casos de prueba consultar **Anexo 5**.

CU5 Proponer Documento

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	título	Campo de texto	No	Se debe especificar al menos una palabra y sin caracteres extraños.
2	autor	Campo de texto	No	Se debe especificar al menos una palabra y sin caracteres extraños.
3	tema	lista de opciones	No	Se debe seleccionar un tema.
4	tipo	lista de opciones	No	Se debe seleccionar un tipo.
5	lenguaje	lista de opciones	No	Se debe seleccionar un lenguaje.
6	file	subir fichero	No	Se debe cargar un archivo.

Tabla 5. Descripción de las variables.

Escenario	Descripción	Variable 1: título	Variable 2: autor	Variable 3: tema	Variable 4: tipo	Variable 5: lenguaje	Variable 6: file	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos.	El usuario introduce todos los datos de manera correcta en el sistema.	V	V	V	V	V	V	El sistema adiciona el documento al sistema y muestra un mensaje: "El documento se a adicionado correctamente".	El usuario llena todos los campos de manera correcta y le da clic en Adicionar o presiona el botón Enter.
EC 1.2	El usuario	I	I	N/A	N/A	N/A	N/A	Si el	El usuario

Capítulo #3 – Implementación y pruebas.

Campo s Vacíos	deja algunos de los campos título o autor sin llenar.							navegador es Firefox a partir de la versión 5 el sistema muestra un mensaje que dice: "Por favor, rellene este campo" o "Please fill out this field" en dependenci a del idioma que tenga el navegador.	deja algunos de los campos título o autor sin llenar y le da clic en Adicionar o presiona el botón Enter.
EC 1.3 No carga el fichero.	El usuario no carga el fichero en la aplicación.	N/A	N/A	N/A	N/A	N/A	I	El sistema muestra un mensaje: "Debe insertar un documento." .	El usuario no introduce el documento que desea publicar y le da clic en Adicionar o presiona el botón Enter.

Tabla 6. Caso de prueba del CU Proponer Documento.

CU19 Registrar Usuario

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	nombre	Campo de texto	No	Se debe especificar al menos una palabra y sin caracteres extraños.
2	email	Campo de texto	No	Se debe especificar un correo con el formato correcto y además valido.
3	username	Campo de texto	No	Se debe especificar una palabra, y solo letras y números, y que no se encuentre en el sistema.
4	plainPassword_first	Campo de texto	No	Se debe especificar una secuencia de caracteres que sea igual a la de la variable 5.
5	plainPassword_second	Campo de texto	No	Se debe especificar una secuencia de caracteres que sea igual a la de la variable 4.

Tabla 7. Descripción de las variables.

Capítulo #3 – Implementación y pruebas.

Escenario	Descripción	Variable 1: nombre	Variable 2: email	Variable 3: username	Variable 4: plainPassword_first	Variable 5: plainPassword_second	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos	El usuario introduce todos los valores de manera correcta en el sistema.	V	V	V	V	V	El sistema adiciona el usuario al sistema y muestra un mensaje informando que se envió al correo un mensaje de confirmación, y el usuario estara desabilitado hasta que se confirme el mismo.	El usuario llena todos los campos de manera correcta y le da clic en Adicionar o presiona el botón Enter.
EC 1.2 Campos Vacíos	El usuario deja todos o algunos campos sin llenar.	I	I	I	I	I	Si el navegador es Firefox a partir de la versión 5 el sistema muestra un mensaje que dice: "Por favor, rellene este campo" o "Please fill out this field" en dependencia del idioma que tenga el navegador.	El usuario deja algun campo de vacío y le da clic en Adicionar o presiona el botón Enter.
EC 1.3 Dato incorrecto.	El usuario introduce un valor incorrecto en el sistema.	V	V	I	V	V	El sistema muestra un mensaje: "El usuario ya existe."	El usuario llena todos los campos de manera correcta pero introduce un usuario que se en cuenta en uso y le da clic en Adicionar o presiona el botón Enter.
		V	I	V	V	V	El sistema muestra un mensaje: "La direccion de correo ya es usada.."	El usuario llena todos los campos de manera correcta pero no coinciden las dos contraseñas que ingresó (variable 4 y 5) y le da clic en Adicionar o presiona el botón Enter.
		V	V	V	V	I	El sistema muestra un mensaje: "La contraseña no coincide".	El usuario llena todos los campos de manera correcta pero no coinciden las dos contraseñas que ingresó (variable 4 y 5) y le da clic en Adicionar o presiona el botón Enter.
		V	V	V	I	V	El sistema muestra un mensaje: "La contraseña no coincide".	El usuario llena los campos pertinentes y da click en la opción buscar o presiona el botón Enter.
		V	V	V	V	V		

Tabla 8. Caso de prueba del CU Registrar Usuario.

Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

Como resultado de la aplicación de este tipo de pruebas se detectaron un total de 23 no conformidades durante la realización de 6 iteraciones distribuidas de la siguiente forma (ver Figura 16). Las no conformidades detectadas se agrupan fundamentalmente en tres tipos de errores: campos con entradas no validadas, mensajes del sistema incorrectos, funcionalidades con deficiencias (ver Figura 17).

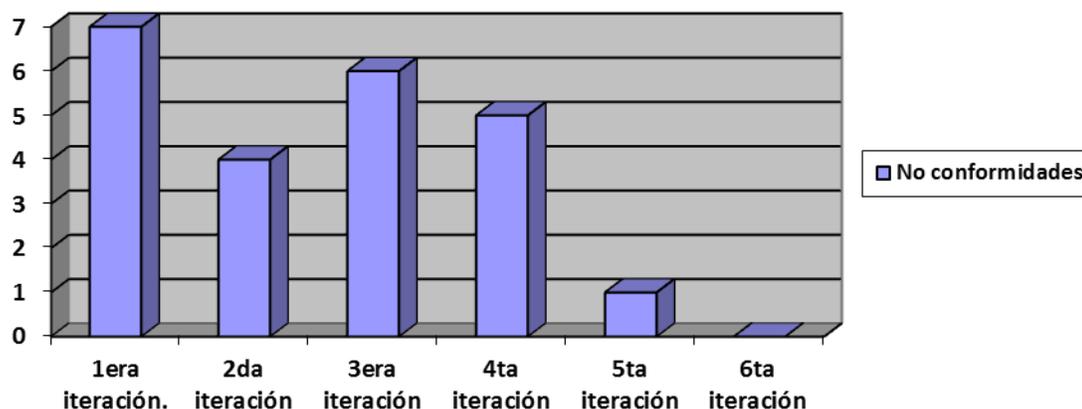


Ilustración 24. No conformidades detectadas distribuidas por iteraciones.

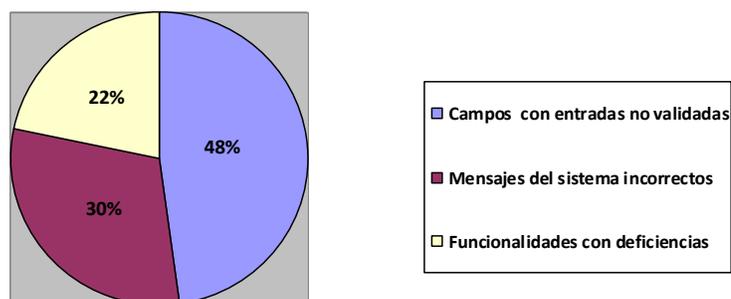


Ilustración 25. Tipos de no conformidades detectadas.

Como se pudo apreciar en los resultados obtenidos al concluir la realización de las pruebas funcionales fueron satisfactorios, ya que las no conformidades encontradas durante su ejecución no constituían deficiencias críticas en la funcionalidad de la aplicación y además todas fueron corregidas satisfactoriamente durante las iteraciones realizadas.

3.4.2. Pruebas unitarias y de integración

Mediante la ejecución de las pruebas funcionales a cada uno de los bundles internos del sistema, de forma tal que se aislaron las partes individuales de la aplicación, fue posible determinar su correcto funcionamiento teniéndose en cuenta el estudio del flujo de datos entre los diferentes componentes que los integran. La ejecución de las pruebas de integración permitió verificar la operación conjunta de cada uno de los componentes del sistema haciendo énfasis en la interacción entre estos, lo que permitió la detección de incoherencias en el funcionamiento de la aplicación. También fue posible la detección de deficiencias en el manejo de entidades dependientes por parte de los componentes ya que se eliminaban referencias y datos que luego eran utilizados por otros elementos del sistema. Todas estas deficiencias

Capítulo #3 – Implementación y pruebas.

detectadas durante la ejecución de estas pruebas se solucionaron de manera inmediata y eficiente, a pesar de que las mismas no representaban un impacto crítico para el funcionamiento de la aplicación.

3.4.3. Pruebas de carga y estrés

Para comprobar la escalabilidad del *software* desarrollado se hace necesario determinar la carga que puede soportar. Para ello se desarrollaron las pruebas de rendimiento utilizando la herramienta Jmeter, lo que arrojó datos importantes acerca del stress que puede soportar el sistema. Se realizó una simulación de carga al sistema con un total de **2060** peticiones enviadas en **20** hilos de ejecución concurrentes, arrojando un **0.0014%** de error al atender dichas peticiones y un tiempo promedio de respuesta de **4.5** segundos (Ver **Anexo 6**). Se debe señalar que la simulación fue realizada en un servidor de bajas prestaciones con las siguientes características: sistema operativo Ubuntu 12.04, microprocesador a 3.0 GHz y una memoria RAM de 2GB, considerablemente inferior a los descrito en los requisitos no funcionales. Los resultados de estas pruebas demuestran que la aplicación desarrollada respondió de manera eficiente, ya que superan a los resultados que se esperaban, a pesar de las bajas prestaciones del servidor donde se encontraba desplegado el sistema para la realización de estas pruebas.

3.4.4. Pruebas de seguridad

Cualquier sistema basado en computadora que maneje información sensible o lleve a cabo acciones que puedan perjudicar (o beneficiar) impropiamente a las personas es un posible objetivo para entradas impropias o ilegales al sistema [32]. Con el objetivo de evaluar la seguridad en un primer nivel, el sistema fue sometido al chequeo de sus elementos por una lista con quince indicadores pertenecientes a cuatro tipos de pruebas de seguridad: pruebas de autorización, pruebas de gestión de sesiones, comprobación del sistema de autenticación y validación de datos. Como resultado de la aplicación de la lista de chequeo fue posible determinar deficiencias en cinco indicadores de impacto crítico en la seguridad. Al ser erradicadas las deficiencias de seguridad detectadas, para comprobar la seguridad en el nivel dos, el sistema fue sometido a un escaneo de vulnerabilidades lo que permitió detectar afectaciones a la seguridad, de estas cuatro de impacto medio en la seguridad del sistema relacionados con la protección contra secuencias de comandos en sitios cruzados (*Cross-site scripting*) y formularios con campos no validados. Luego de resueltas las deficiencias encontradas el sistema fue sometido nuevamente a un escaneo de vulnerabilidades para garantizar que habían sido resueltas, arrojando como resultado la inexistencia de errores de seguridad en la aplicación.

3.5. Conclusiones

Se detalló el estándar de codificación utilizado, logrando una mayor legibilidad y limpieza del código. Además se construyó el sistema de acuerdo a lo definido durante el diseño de cada componente. La realización de las pruebas funcionales, unitarias, de integración y de rendimiento permitieron la detección de errores y deficiencias que fueron corregidos, permitiendo validar el correcto funcionamiento del *software* atendiendo a los resultados obtenidos.

CONCLUSIONES GENERALES

A partir del estudio de los procesos de publicación y recuperación de documentos digitales en aplicaciones web, se realizó un análisis del estado del arte sobre diferentes sistemas encargados de la ejecución de estos procesos, que permitió determinar un grupo de funcionalidades y tendencias que facilitaron la elaboración de la propuesta de solución. Después del estudio del arte se comprobó que ninguno de los sistemas estudiados resolvía totalmente la problemática planteada por lo que a partir de las deficiencias detectadas se definieron un grupo de requisitos que derivaron, como principal resultado, en un sistema informático que permite la publicación y recuperación de documentos digitales. Esta aplicación puede llegar a ser una importante herramienta para aumentar la disponibilidad de documentos digitales en la intranet nacional. A la aplicación le fueron realizadas varias pruebas, entre las que se encuentran las pruebas funcionalidad, seguridad, carga y estrés e integración obteniendo resultados satisfactorios lo que permitió validar el correcto funcionamiento del sistema propuesto. Por lo que se consideran cumplidos los objetivos propuestos al inicio de la investigación.

RECOMENDACIONES

- Realizar mejoras por parte de especialistas en diseño a la apariencia visual y las interfaces de usuario de la aplicación propuesta.
- Integrar la aplicación propuesta al motor de búsqueda Orión.
- Desplegar la Plataforma Colaborativa para la Publicación de Documentos Digitales en la Universidad de las Ciencias Informáticas como paso previo a su despliegue en la intranet cubana.

REFERENCIAS BIBLIOGRÁFICAS

1. Dirección de Investigaciones de la Universidad de las Ciencias Informáticas. Balance de Ciencia e Innovación Tecnológica 2011, 2012. [Disponible en: http://investigaciones.uci.cu/files/Descargas/Balance_CTI/Informe_balance_2011.pdf].
2. Martínez Comeche, Juan A. Teoría de la información documental y de las instituciones documentales. Madrid, Síntesis, 1995.
3. Codina, Lluís. El llibre digital. Barcelona, Generalitat de Catalunya, 1996.
4. Róbaldo de Robles ML. Revistas científicas electrónicas. Publicaciones electrónicas. Revista de la Dirección General de Bibliotecas de la UNAM [publicación periódica en línea] 1998 Jul-Dic [citada 2001 Dic.]; Disponible en: <http://www.dgbiblio.unam.mx/servicios/dgb/publicdgb/bole/fulltext/vo/12/electronicas.html>.
5. Publicaciones Electrónicas. [Sitio en Internet] Disponible en: URL <http://est.unileon.es/medicina/temas/5.htm#publicaciones>. [Consultado:6 de diciembre 2001.].
6. Michel López Camino, Raidel Muñoz Vidal. Gestor de recuperaciones dinámicas para aplicaciones web. ISPJAE, 2006.
7. Witten, I. H. Moffat A. and Bell T. C. Compressing and Indexing Documents and Images. Second Edition ed. s.l. : Morgan Kaufmann, 1999.
8. G. Figueroa, Roberth, J. Solís, Camilo y A. Cabrera, Armando. METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. *Universidad Técnica Particular de Loja, Escuela de Referencias bibliográficas Ciencias en Computación*. [En línea] [Citado el: 16 de noviembre de 2012.] [Disponible en: <http://www.scribd.com/doc/42572078/Articulo-Metodologia-de-Sw-Formato>.]
9. HENRIK, K., SCRUM y XP desde las trincheras. C4Media. 2007. ISBN: 978-1-4303-2264-1. Número de páginas 122.
10. GREGORIO, R. M., JORGE, F. Z., Programación extrema y Software Libre, 2002.
11. Ricardo Balduino. Introduction to OpenUP (Open Unified Process) [En línea] [Citado el: 16 de noviembre de 2012.] [Disponible en: <http://www.eclipse.org/epf/general/OpenUP.pdf>.]
12. Hernández Orallo, Enrique. Universidad Politécnica de Valencia DISCA. [En línea] [Citado el: 16 de noviembre de 2012.] [Disponible en: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.]

Referencias Bibliográficas

13. Antonio Moreno y Villena. HTML historia y características generales. [En línea] [Citado el: 17 de noviembre de 2012.] [Disponible en: http://www.aeemt.com/contenidos_socios/Informatica/Informac_Informat_Tecnolog/AMV_AGI_AEE_MT_HTML_Historia.pdf.]
14. INTECO - Instituto Nacional de Tecnologías de la comunicación. Centro de Referencia en Accesibilidad y Estándares Web. [En línea] [Citado el: 17 de noviembre de 2012.] [Disponible en: http://www.inteco.es/file/Kilat7eU_6hCnkU_DZrdhA.]
15. Javascript. librosweb.es. [En línea] [Citado el: 17 de noviembre de 2012.] [Disponible en: <http://www.librosweb.es/javascript/capitulo1.html>.]
16. PHP. [En línea] [Citado el: 17 de noviembre de 2012.] [Disponible en: <http://www.php.net/manual/es/intro-what-is.php>.]
17. W3C.SOAP Version 1.2 Pa 1: Messaging Framework. [En línea] W3C Consortium, Abril de 2007. [Disponible en: <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/#encapsulation>.]
18. Eguiluz, Javier. Desarrollo web ágil con Symfony2. Publicaciones easybook, 2011. Número de páginas 573.
19. Martínez, Rafael. Principal | www.postgresql-es.org. [En línea] [Citado el: 18 de noviembre de 2012.] <http://www.postgresql-es.org/principal>.
20. 10gen. MongoDB. [En línea] 2011. [Citado el: 18 de noviembre de 2012.] [Disponible en: <http://www.mongodb.com>.]
21. Kabir Mohammed J. *La biblia de Servidor Apache2*. 2004.
22. Sitio de descargas de software. [En línea] [Citado el: 18 de noviembre de 2012.] [Disponible en: http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.]
23. Netbeans.org. Sitio oficial NetBeans.org [En línea] [Citado el: 18 de noviembre de 2012.] [Disponible en: <http://netbeans.org/features/php/index.html>.]
24. Pressman, Roger S. Ingeniería de Software un enfoque práctico. Naucalpan de Juárez. Prentice Hall, 1999. Número de páginas 536.
25. Craig Larman. UML y Patrones 2da Edición, Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Mc Graw Hill. Número de páginas 614.

Referencias Bibliográficas

26. Gregory Raghavan, Ford Sridhar, Gary Zelesnik. Lecture Notes on Requirements Elicitation.
27. Fowler, M. *Analysis Patterns: Reusable Object Models*. Reading. Addison-Wesley, 1996.
28. GUNNAR ÖVERGAARD, K. P. Use Cases Patterns and Blueprints, Addison Wesley Professional, 2004.
29. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. Pattern – Oriented Software Architecture. A System of Patterns. John Wiley & Sons, 1991.
30. Microsoft. Microsoft Developer Network. [En línea] 2013. [Citado el: 13 de Mayo de 2013.] [Disponible en: <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.]
31. Martin, J. y Odell, J.. *Object-Oriented Methods: A Foundation*. Englewood Cliffs. Prentice-Hall, 1995.
32. B, Ing. Alexander Oré. 2008. Calidady Software.com. *Calidady Software.com*. [En línea] Nazcasoft.com, 2008. [Citado el: 2 de 4 de 2013.] <http://www.calidadyssoftware.com/testing.php>.
33. Bass, L., Clements, P., & Kazman, R. Software Architecture in practice. Addison-Wesley, 1998.

BIBLIOGRAFÍA

- **Chodorow, Kristina. Dirolf, Michael.** MongoDB: The Definitive Guide. O'Reilly Media, Sept 2010. **ISBN: 978-1-449-38156-1.**
- **Javier Eguiluz**, Desarrollo web ágil con Symfony2, 2011. Publicaciones **easybook** versión **4.8-DEV**. Número de páginas:573.
- **(UCI) Dpto. Soluciones Informáticas para Internet, Centro de Ideoinformática.** Trabajo con MongoDB desde Symfony2, marzo 2012.
- **Pressman, Roger S.** Ingeniería de Software un enfoque práctico. Naucalpan de Juárez. Prentice Hall, 1999. Número de páginas 536.
- **Potencier Fabien, Zanionotto Francois.** Symfony la Guía definitiva. Librosweb, 2007. Número de páginas: 435.
- **Hernández León, Rolando Alfredo, Coello González Sayda.** El proceso de la investigación científica, 2011.
- **INTECO, Laboratorio Nacional de Calidad de Software.** Ingeniería del Software: Metodologías y Ciclos de Vida. España : s.n., 2009.
- **Ideoinformática, Centro.** Configuración de la metodología OpenUP. Ciudad de La Habana. Cuba : s.n., 2012.
- **Jacobson, Ivar.** El proceso unificado de desarrollo de software. Madrid : Addison Wesley, 2000.
- **Peña, Ing. Yadira Machado.** Planificación, Diseño y Ejecución de Pruebas Funcionales. Habana: Calisoft, marzo-2011.
- **B, Ing. Alexander Oré. 2008.** Calidad Software.com. *Calidad Software.com*. [En línea] Nazcasoft.com, 2008. [Citado el: 2 de 4 de 2013.] <http://www.calidadyssoftware.com/testing.php>.
- **PHP**, 2013. [Disponible en: <http://www.php.net/manual/es/intro-what-is.php>.]

GLOSARIO DE TÉRMINOS

Bundle: Son una mezcla entre los módulos y los *plugins*. Es aquí donde se encuentra la programación realmente y la idea del Bundle es que si se necesita una misma funcionalidad dentro de otro proyecto debería poder copiarlo completo al nuevo proyecto y debería ser lo suficientemente independiente como para funcionar.

CASE: Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador), ingeniería basada en aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Hipertexto: es una tecnología que organiza una base de información en bloques distintos de contenidos, conectados a través de una serie de enlaces cuya activación o selección provoca la recuperación de información.

HTTP El protocolo de transferencia de hipertexto o *HyperText Transfer Protocol* es el protocolo usado en cada transacción web. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

ODM: Acrónimo de *Object-Document Mapping* (Mapeo Objeto-Documento). Es empleado por el marco de trabajo mediante la utilización de técnicas de programación para convertir datos a objetos y viceversa, permitiendo el trabajo con datos persistentes como si formaran parte de una base de datos orientada a objetos.

Servidor: en informática, un servidor es un tipo de *software* que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al computador en el cual funciona ese *software*, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Software: se refiere al equipamiento lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.

ANEXOS

Anexo 1

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF1.	Registrar usuario.	El sistema debe permitir que se introduzcan los datos de un usuario que se quiera registrar.	Media	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario introduzca todos los datos.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF2.	Cambiar Contraseña.	El sistema debe permitir que un usuario cambie su contraseña cuando el mismo lo requiera.	Media	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario se encuentre registrado y autenticado en el sistema.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
----	--------	-------------	-------------	------------------------

Anexos

RF3.	Eliminar usuario.	El sistema debe permitir eliminar un usuario del sistema.	Media	Alta
Prototipo				
Campos	Tipos de Datos		Reglas o Restricciones	
	No procede		Que esta operación la realice un usuario con privilegios de administración.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF4.	Autenticar usuario.	El sistema debe permitir que un usuario se autentique en el mismo.	Media	Alta
Prototipo				
Campos	Tipos de Datos		Reglas o Restricciones	
	No procede		Que el usuario se encuentre registrado en el sistema.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF5.	Cerrar sesión de usuario.	El sistema debe permitir finalizar la sesión del usuario.	Media	Alta
Prototipo				
Campos	Tipos de Datos		Reglas o Restricciones	

Anexos

	No procede	Que el usuario se encuentre autenticado en el sistema.
Observaciones		

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF6.	Modificar permisos de usuario.	El sistema debe permitir modificar los privilegios de un usuario en el sistema.	Media	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario que realice esta operación posea los privilegios suficientes.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF7.	Bloquear usuario.	El sistema debe permitir bloquear todas las operaciones que realice un usuario que haya tenido un comportamiento inadecuado en el sistema.	Media	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario que realice esta operación posea los privilegios suficientes.	

Anexos

Observaciones	
----------------------	--

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF8.	Proponer documento.	El sistema debe permitir subir un documento digital y los datos asociados a este.	Alta	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario que realice esta operación posea los privilegios suficientes y el documento tenga algunos de los formatos como: pdf, doc, txt.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF9.	Modificar datos de documento.	El sistema debe permitir modificar los datos asociados a un documento.	Media	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario que realice esta operación posea los privilegios suficientes.	
Observaciones				

Anexos

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF10.	Eliminar documento.	El sistema debe permitir eliminar un documento.	Media	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario que realice esta operación posea los privilegios suficientes.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF11.	Descargar documento.	El sistema debe permitir descargar un documento.	Baja	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario que realice esta operación posea los privilegios suficientes.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad
----	--------	-------------	-------------	-----------

Anexos

				para cliente
RF12.	Comentar documento.	El sistema debe permitir al usuario realizar comentarios acerca de algún documento.	Media	Media
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario se encuentre autenticado en el sistema.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF13.	Eliminar comentario.	El sistema debe permitir eliminar un comentario no deseado.	Media	Media
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario que realice esta operación posea los privilegios suficientes.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF14.	Compartir documento.	El sistema debe permitir que los documentos publicados se puedan	Alta	Baja

Anexos

		compartir en las redes sociales.		
Prototipo				
Campos	Tipos de Datos		Reglas o Restricciones	
	No procede		Que el usuario se encuentre autenticado en el sistema.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF15.	Votar documento.	El sistema debe permitir que un usuario vote por los documentos de su preferencia.	Media	Baja
Prototipo				
Campos	Tipos de Datos		Reglas o Restricciones	
	No procede		Que el usuario se encuentre autenticado en el sistema.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF16.	Me gusta documento.	El sistema debe permitir que un usuario comparta su criterio en cuanto a si le gusta un determinado documento.	Media	Baja
Prototipo				

Anexos

Campos	Tipos de Datos	Reglas o Restricciones
	No procede	Que el usuario se encuentre autenticado en el sistema.
Observaciones		

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF17.	Seguir documento.	El sistema debe ser capaz de permitir que un usuario se haga seguidor de un documento, y entonces el sistema tiene que enviar reportes al correo de este usuario, informando de lo que suceda relacionado con ese documento.	Media	Media
Prototipo				
Campos	Tipos de Datos	Reglas o Restricciones		
	No procede	Que el usuario se encuentre autenticado en el sistema.		
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF18.	Publicar documento.	El sistema debe permitir que un usuario autorizado apruebe o no, si un documento se puede publicar en el sistema.	Media	Alta
Prototipo				
Campos	Tipos de Datos	Reglas o Restricciones		
	No procede	Que el usuario que realice esta operación posea los		

Anexos

		privilegios suficientes.
	Observaciones	

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF19.	Seguir tema.	El sistema debe ser capaz de permitir que un usuario se haga seguidor de una categoría, y entonces el sistema tiene que enviar reportes al correo de este usuario, informando de los nuevos documentos que se publiquen en esta categoría.	Media	Media
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario se encuentre autenticado en el sistema.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF20.	Realizar búsquedas básicas.	El sistema debe permitir al usuario según el criterio de búsqueda realizar la misma en los títulos de los documentos, y luego mostrar el resultado de esta.	Media	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario se encuentre autenticado en el sistema.	

Anexos

Observaciones	
----------------------	--

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF21.	Realizar búsquedas avanzadas.	El sistema debe permitir al usuario según el criterio de búsqueda y el filtrado según los datos asociados a los documentos, realizar la búsqueda, y luego mostrar el resultado de la misma.	Alta	Media
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario se encuentre autenticado en el sistema.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF22.	Mostrar información de un documento.	El sistema debe permitir mostrar a los usuarios todos los datos asociados a un documento.	Baja	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario se encuentre autenticado en el sistema.	
Observaciones				

Anexos

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF23.	Confirmar Usuario.	El sistema debe permitir confirmar la identidad de los usuarios.	Media	Alta
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	No procede	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF24.	Dejar de seguir documento.	El sistema debe permitir a solicitud del usuario dejar de seguir un documento, donde se deje de enviar reportes de actividad sobre el documento al correo del mismo.	Media	Media
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario que realice esta operación se encuentre autenticado y sea seguidor del documento.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
----	--------	-------------	-------------	------------------------

Anexos

RF25.	Dejar de seguir tema.	El sistema debe permitir a solicitud del usuario dejar de seguir un documento, donde se deje de enviar reportes de actividad sobre el tema al correo del mismo.	Media	Media
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario que realice esta operación se encuentre autenticado y sea seguidor del tema.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF26.	Adicionar tema.	El sistema debe permitir a los administradores adicionar un nuevo tema a la aplicación.	Media	Media
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que esta operación la realice un usuario con privilegios de administración.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
----	--------	-------------	-------------	------------------------

Anexos

RF27.	Eliminar tema.	El sistema debe permitir a los administradores eliminar un tema de la aplicación.	Media	Media
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que esta operación la realice un usuario con privilegios de administración.	
Observaciones				

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF28.	Enviar reporte.	El sistema debe permitir el envío de reportes de actividades acerca de un tema o documento a través del correo a los usuarios que sean seguidores de algún tema o documento.	Alta	Media
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que esta operación se realice con los usuarios que soliciten ser seguidores de algún tema o documento.	
Observaciones				

Anexos

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF29.	Buscar Usuario.	El sistema debe permitir realizar búsqueda de los usuarios registrados en el sistema.	Media	Baja
Prototipo				
Campos		Tipos de Datos	Reglas o Restricciones	
		No procede	Que el usuario que realice esta operación posea los privilegios suficientes.	
Observaciones				

Anexo 2

CU 1. Realizar búsqueda básica.

Objetivo	Realizar búsqueda básica de documentos en la base de datos del sistema.
Actores	Usuario anónimo.
Resumen	El CU inicia cuando un usuario desea realizar una búsqueda, el sistema muestra la interfaz correspondiente con la opción Buscar, finaliza cuando se muestran los resultados.
Complejidad	Media
Prioridad	Alta
Precondiciones	Que el usuario introduzca un criterio de búsqueda
Postcondiciones	Se muestra la información de acuerdo al criterio de búsqueda introducido.
Flujo de eventos	

Anexos

Flujo básico realizar búsqueda		
	Actor	Sistema
	Indica el criterio de búsqueda en el navegador y oprime el botón Buscar.	
		Si el criterio de búsqueda es correcto muestra los resultados de forma ordenada de acuerdo a la relevancia de los mismos. Estos resultados muestran a su vez una serie de información como son: título, autor, tema, tipo de documento y el tamaño del archivo.
		Muestra el número total de resultados encontrados.
		Muestra las palabras introducidas en el criterio de búsqueda de forma resaltada en los resultados.
		Muestra un paginado cuando se encuentren varios resultados que no se puedan ver todos en la primera página, lo que le permite al usuario acceder a otras páginas de una forma más rápida y eficiente.
		Termina el CU.
Flujos alternos		
Relaciones		

Anexos

Requisitos no funcionales		
Asuntos pendientes		

CU 2. Descargar documento.

Objetivo	Permitir descargar los documentos con que cuenta el sistema.	
Actores	Usuario anónimo.	
Resumen	El CU inicia cuando el usuario desea descargar un documento y accede al mismo, la aplicación le muestra una opción para hacerlo y finaliza cuando se realiza la descarga.	
Complejidad	Baja	
Prioridad	Alta	
Precondiciones		
Postcondiciones		
Flujo de eventos		
Flujo básico completar criterio de búsqueda		
	Actor	Sistema
	Accede la interfaz de un documento.	
2.		Muestra los datos de un documento y entre las opciones que se muestran se encuentra la de descargar documento.
3.	Selecciona la opción descargar documento.	
4.		Descarga el archivo correspondiente al documento.
5.		Termina el CU.

Anexos

Flujos alternos		
Nº Evento <Condición que dio lugar a la extensión>		
	Actor	Sistema
Relaciones	CU Incluidos	
	CU Extendidos	Mostrar información de un documento.
Requisitos no funcionales		
Asuntos pendientes		

CU 3. Autenticar usuario.

Objetivo	Autenticar al usuario en el sistema.	
Actores	Usuario anónimo.	
Resumen	El caso de uso inicia cuando el usuario desea autenticarse y accede a la interfaz para realizar esta operación, finaliza cuando el usuario se ha autenticado correctamente.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones		
Postcondiciones	El usuario se autentica por lo que deja de ser un usuario anónimo.	
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Accede a esta opción.	
2.		Muestra la interfaz para realizar esta operación que tiene dos campos

Anexos

		usuario y contraseña.
3.	Introduce el usuario y la contraseña en la interfaz y los envía a la aplicación.	
4.		Comprueba que los datos introducidos por el usuario sean correctos.
5.		Direcciona al usuario hacia la interfaz principal del sistema.
3.		Termina el CU.
Flujos alternos		
4. Los datos introducidos por el usuario no tienen coincidencias en el sistema		
	Actor	Sistema
5.		Muestra un mensaje de error donde se le comunica al usuario que no existen el par usuario y contraseña suministrados por el en la base de datos del sistema.
2.		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 4. Realizar búsqueda avanzada.

Objetivo	Realizar búsquedas avanzadas de información específica en la base de datos del sistema.
Actores	Usuario anónimo.

Anexos

Resumen	El CU inicia cuando un usuario desea realizar una búsqueda avanzada, el sistema muestra la interfaz correspondiente con la opción Búsqueda Avanzada, el usuario realiza la búsqueda y se muestran los resultados obtenidos finaliza así el CU.	
Complejidad	Alta	
Prioridad	Media	
Precondiciones		
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción Búsqueda Avanzada.	
2.		Muestra la interfaz correspondiente con los diferentes criterios por los cuales es posible realizar la búsqueda.
3.		Permite restringir la búsqueda de acuerdo a diferentes criterios como son: Idioma (El sistema debe mostrar los resultados de acuerdo con el idioma que elija el usuario.), Tema (El sistema debe buscar documentos referentes a un tema en específico), Tipo (El sistema debe realizar búsquedas sobre los documentos de un tipo determinado), Formato (El sistema debe buscar documentos que se encuentran en un formato determinado), Autor (La aplicación debe mostrar los documentos de un autor determinado).
4.	Accede a la acción Buscar.	

Anexos

5.		Muestra los resultados en un listado paginado de acuerdo a la configuración personal del usuario.
6.		Termina el CU.
Flujos alternos		
Nº Evento <Condición que dio lugar a la extensión>		
	Actor	Sistema
7.		
8.		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 6. Comentar documento.

Objetivo	Permitir a los usuarios del sistema emitir comentarios sobre los documentos.
Actores	Usuario autenticado
Resumen	El CU se inicia cuando un usuario desea comentar un documento de los publicados en el sistema.
Complejidad	Media
Prioridad	Media
Precondiciones	El usuario debe estar autenticado.
Postcondiciones	

Anexos

Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona el documento que desea comentar.	
2.		Muestra información detallada del documento entre la que se encuentra los comentarios del mismo, además de la opción comentar documento.
3.	Selecciona la opción añadir comentario.	
4.		Muestra un campo donde el usuario puede introducir su comentario.
5.	Introduce su comentario y selecciona publicar.	
6.		Se añade el comentario y se muestra el comentario emitido en los comentarios del documento.
7.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Mostrar información de un documento
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Anexos

CU 7. Compartir documento.

Objetivo	Compartir un documento en las redes sociales.	
Actores	Usuario autenticado	
Resumen	El CU se inicia cuando un usuario que se encuentra autenticado en el sistema desea compartir un documento en las redes sociales.	
Complejidad	Alta	
Prioridad	Baja	
Precondiciones	El usuario debe estar autenticado.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción compartir documento en una red social.	
2.		Muestra una interfaz que el usuario se autentique en la red social que seleccionó.
3.	Escribe los datos para autenticarse en la red social.	
4.		Comparte el documento en el perfil del usuario en la red social.
5.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Mostrar información de un documento
	CU	

Anexos

	Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 8. Votar documento.

Objetivo	Permitir al usuario votar por un documento.	
Actores	Usuario autenticado	
Resumen	El CU se inicia cuando el usuario desee votar por un documento.	
Complejidad	Media	
Prioridad	Baja	
Precondiciones	El usuario debe estar autenticado.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona un documento.	
2.		Muestra la información detallada del documento.
3.	Selecciona en el campo que se muestra para votar por un documento.	
4.		Registra el voto efectuado por el usuario.
5.		Termina el CU.
Flujos alternos		

Anexos

		Actor	Sistema
Relaciones	CU Incluidos	Mostrar información de un documento	
	CU Extendidos		
Requisitos no funcionales			
Asuntos pendientes			

CU 9. Me gusta documento.

Objetivo	Permitir al usuario expresar que un documento le gusta.		
Actores	Usuario autenticado		
Resumen	El CU se inicia cuando el usuario accede a los datos de un documento y presiona en la opción Me gusta .		
Complejidad	Media		
Prioridad	Baja		
Precondiciones	El usuario debe estar autenticado.		
Postcondiciones			
Flujo de eventos			
Flujo básico <Nombre del flujo básico>			
	Actor	Sistema	
1.	Selecciona un documento.		
2.		Muestra la información detallada del documento.	
3.	Selecciona en la opción Me gusta que se muestra para un documento.		

Anexos

4.		Registra la acción del usuario en el sistema.
5.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Mostrar información de un documento
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 10. Seguir documento.

Objetivo	Permitir a los usuarios del sistema seguir las actividades que se desempeñen sobre un documento determinado.
Actores	Usuario autenticado
Resumen	El CU se inicia cuando un usuario que se encuentra autenticado en el sistema desea seguir los comentarios y trayectoria de un documento determinado y se registra como seguidor de un documento.
Complejidad	Media
Prioridad	Media
Precondiciones	El usuario debe estar autenticado.
Postcondiciones	El sistema envía reportes sobre las actividades que se realicen sobre el documento seguido a la dirección de correo electrónico del usuario.

Anexos

Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona un documento.	
2.		Muestra la información detallada del documento.
3.	Selecciona en la opción Seguir que se muestra para un documento.	
4.		Añade el usuario a la lista de seguidores del documento.
6.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Mostrar información de un documento, enviar reporte.
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 11. Seguir tema.

Objetivo	Permitir a los usuarios del sistema seguir las actividades que se desempeñen sobre un tema determinado.
Actores	Usuario autenticado
Resumen	El CU se inicia cuando un usuario que se encuentra autenticado en el sistema desea seguir los comentarios y documentos que se

Anexos

	encuentran sobre un tema determinado y se registra como seguidor de un tema.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado.	
Postcondiciones	El sistema envía reportes sobre las actividades que se realicen sobre el tema seguido a la dirección de correo electrónico del usuario.	
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona un tema.	
2.		Muestra la información detallada del tema.
3.	Selecciona en la opción Seguir que se muestra para un tema.	
4.		Añade el usuario a la lista de seguidores del tema.
6.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Enviar reporte.
	CU Extendidos	
Requisitos no funcionales		
Asuntos		

Anexos

pendientes	
-------------------	--

CU 12. Modificar permisos de usuario.

Objetivo	Permitir modificar los permisos de un usuario	
Actores	Super administrador	
Resumen	El CU inicia cuando el super administrador selecciona la opción gestionar usuarios en el panel de administración del sistema, busca el usuario que desea cambiar y actualiza los permisos del mismo.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado y ser super administrador del sistema.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción gestionar usuarios en el panel de administración del sistema.	
2.		Muestra una interfaz donde se muestran todos los usuarios y que además permite realizar búsquedas de usuarios sobre la base de datos.
3.	Accede al usuario que desea modificar y selecciona la opción modificar permisos.	
4.		Carga los permisos del usuario en una interfaz que permite editar los mismos.
5.	Realiza los cambios en los permisos del usuario seleccionado y selecciona la opción aceptar.	

Anexos

7.		Registra los cambios realizados por el administrador.
8.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Buscar usuario
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 13. Bloquear usuario.

Objetivo	Permitir a un administrador del sistema bloquear el acceso a un usuario.
Actores	Administrador
Resumen	El CU el administrador selecciona la opción gestionar usuarios en el panel de administración del sistema, busca el usuario que desea bloquear y selecciona en la opción que permite realizar esta operación.
Complejidad	Media
Prioridad	Alta
Precondiciones	El usuario debe estar autenticado.
Postcondiciones	
Flujo de eventos	

Anexos

Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción gestionar usuarios en el panel de administración del sistema.	
2.		Muestra una interfaz donde se muestran todos los usuarios y que además permite realizar búsquedas de usuarios sobre la base de datos.
3.	Accede al usuario que desea modificar y selecciona la opción bloquear.	
4.		Registra los cambios realizados por el administrador.
5.		Finaliza el CU
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Buscar usuario
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 14. Publicar documento.

Objetivo	Hacer público un documento en el sistema.
Actores	Administrador
Resumen	El CU se inicia cuando el administrador decide hacer público un

Anexos

	documento que ha sido proporcionado por un usuario, busca el documento revisa sus datos y contenido y lo publica.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones		
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción gestionar documentos en el panel de administración del sistema.	
2.		Muestra una interfaz donde se muestran todos los documentos y que además permite realizar búsquedas de documentos sobre la base de datos.
3.	Accede al documento que desea publicar y selecciona la opción publicar.	
4.		
5.		Finaliza el CU.
Flujos alternos		
	Actor	Sistema
6.		
Relaciones	CU Incluidos	Buscar documento
	CU Extendidos	
Requisitos no funcionales		

Anexos

Asuntos pendientes	
---------------------------	--

CU 15. Eliminar comentario.

Objetivo	Eliminar un comentario del sistema.	
Actores	Administrador	
Resumen	El caso de uso se inicia cuando el administrador del sistema desea eliminar un comentario, el mismo accede a los comentarios que existen sobre un documento determinado, y donde se muestra el comentario y lo elimina.	
Complejidad	Media	
Prioridad	Media	
Precondiciones		
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona el documento al que pertenece el comentario.	
2.		Muestra la información detallada del documento entre ella los comentarios emitidos sobre el mismo.
3.	Selecciona en el documento el comentario a eliminar y en él la opción eliminar.	
4.		Elimina el comentario de la base de datos.
5.		Actualiza la página con la información del documento.
6.		Termina el CU.

Anexos

Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Mostrar información de documento, buscar documento.
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 16. Eliminar usuario.

Objetivo	Eliminar un usuario de los que se encuentran registrados en el sistema.	
Actores	Super administrador	
Resumen	El CU se inicia cuando el super administrador del sistema requiere eliminar un usuario, el mismo busca el usuario que desea eliminar y accede a la opción eliminar y el sistema los elimina de los registros.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado y ser super administrador del sistema.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema

Anexos

1.	Selecciona la opción gestionar usuarios en el panel de administración del sistema.	
2.		Muestra una interfaz donde se muestran todos los usuarios y que además permite realizar búsquedas de usuarios sobre la base de datos.
3.	Accede al usuario que desea eliminar y selecciona la opción eliminar.	
4.		Registra los cambios realizados por el administrador.
5.		Finaliza el CU
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Buscar usuario
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 17. Mostrar información de un documento.

Objetivo	Mostrar la toda la información que se tiene de un documento, así como los comentarios que han emitido los usuarios sobre el mismo.
Actores	Usuario anónimo
Resumen	El CU se inicia cuando un usuario anónimo accede a un documento y el sistema le muestra toda la información con que se cuenta de ese

Anexos

	documento.	
Complejidad	Baja	
Prioridad	Alta	
Precondiciones		
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Accede a la opción que le muestra toda la información del documento.	
2.		Muestra una interfaz donde se muestra la información detallada del documento.
3.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 18. Enviar reporte.

Objetivo	Enviar reportes al correo de un usuario.
-----------------	--

Anexos

Actores	Usuario autenticado.	
Resumen	El CU se inicia cuando un usuario que se encuentra autenticado en el sistema desea seguir un documento o un tema, el sistema una vez a la semana le envía al usuario un correo con toda la actividad realizada sobre los documentos o temas que está siguiendo un usuario.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario debe de haber seleccionado seguir un tema o un documento.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.		Genera un correo para cada usuario que sigue un tema o un documento.
2.		Envía los correos generados.
3.		Termina el CU.
Flujos alternos		
4. El archivo proporcionado por el usuario no cumple con las restricciones definidas por el sistema.		
	Actor	Sistema
6.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		

Anexos

Asuntos pendientes	
---------------------------	--

CU 20. Confirmar usuario.

Objetivo	Habilitar el usuario como válido en el sistema.	
Actores	Usuario anónimo	
Resumen	El CU se inicia cuando el usuario accede desde su cuenta de correo al enlace que le ha sido enviado por el sistema, este registra el acceso desde la cuenta de correo y habilita al usuario.	
Complejidad	Media	
Prioridad	Media	
Precondiciones		
Postcondiciones	El usuario ya puede acceder al sistema.	
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Accede desde su cuenta de correo al enlace que le ha sido enviado por el sistema.	
2.		Autentica al usuario por el acceso a la dirección del enlace.
3.		Se habilita el usuario en el sistema.
4.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	

Anexos

	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 21. Bloquear documento.

Objetivo	Permitir a un administrador del sistema bloquear el acceso a un documento.	
Actores	Administrador	
Resumen	El CU el administrador selecciona la opción gestionar documentos en el panel de administración del sistema, busca el documentos que desea bloquear y selecciona en la opción que permite realizar esta operación.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado y ser administrador del sistema.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción gestionar documentos en el panel de administración del sistema.	
2.		Muestra una interfaz donde se muestran todos los documentos y que además permite realizar búsquedas de documentos sobre la base de datos.
3.	Accede al documentos que desea modificar	

Anexos

	y selecciona la opción bloquear.	
4.		Registra los cambios realizados por el administrador.
5.		Finaliza el CU
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Buscar documentos
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 22. Modificar datos de un documento.

Objetivo	Permitir modificar los datos de un documento.
Actores	Administrador
Resumen	El CU el administrador selecciona la opción gestionar documentos en el panel de administración del sistema, busca el documento que desea cambiar y actualiza los datos del mismo.
Complejidad	Media
Prioridad	Alta
Precondiciones	El usuario debe estar autenticado y ser administrador del sistema.
Postcondiciones	
Flujo de eventos	
Flujo básico <Nombre del flujo básico>	

Anexos

	Actor	Sistema
1.	Selecciona la opción gestionar documentos en el panel de administración del sistema.	
2.		Muestra una interfaz donde se muestran todos los documentos y que además permite realizar búsquedas de documentos sobre la base de datos.
3.	Accede al documento al que desea cambiarle los datos y selecciona la opción modificar.	
4.		Carga los datos del documento en una interfaz que permite editar los mismos.
5.	Realiza los cambios en los datos del documento seleccionado y selecciona la opción aceptar.	
7.		Registra los cambios realizados por el administrador.
8.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Buscar documento
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Anexos

CU 23. Eliminar documento.

Objetivo	Eliminar un documento de los que se encuentran registrados en el sistema.	
Actores	Administrador	
Resumen	El CU se inicia cuando el administrador del sistema requiere eliminar un documento, el mismo busca el documento que desea eliminar accede a la opción eliminar y el sistema los elimina de los registros.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado y ser administrador del sistema.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción gestionar documentos en el panel de administración del sistema.	
2.		Muestra una interfaz donde se muestran todos los documentos y que además permite realizar búsquedas de documentos sobre la base de datos.
3.	Accede al documento que desea eliminar y selecciona la opción eliminar.	
4.		Registra los cambios realizados por el administrador.
5.		Finaliza el CU
Flujos alternos		

Anexos

	Actor	Sistema
Relaciones	CU Incluidos	Buscar documento
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 24. Buscar documento.

Objetivo	Buscar un documento por el título en el sistema.	
Actores	Administrador.	
Resumen	El CU se inicia cuando el administrador necesita encontrar un documento en el sistema el mismo introduce un texto y el sistema lo busca en el título de los documentos y devuelve las coincidencias encontradas.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado y ser administrador del sistema.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Introduce el texto a buscar y selecciona la opción buscar.	
2.		Busca coincidencias en los títulos de los documentos.

Anexos

3.		Muestra los resultados obtenidos.
4.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 25. Buscar usuario.

Objetivo	Buscar un usuario en el sistema.	
Actores	Administrador.	
Resumen	El CU se inicia cuando el administrador necesita encontrar un usuario en el sistema el mismo introduce un texto y el sistema lo busca en el usuario de las personas registradas en el sistema.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado y ser administrador del sistema.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema

Anexos

1.	El administrador introduce el usuario a buscar y selecciona la opción buscar.	
2.		El sistema busca coincidencias en los usuarios
3.		El sistema muestra los resultados obtenidos.
4.		Termina el CU.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 26. Cambiar contraseña.

Objetivo	Permitir cambiar la contraseña de un usuario
Actores	Usuario autenticado
Resumen	El CU inicia cuando el usuario selecciona la opción cambiar la contraseña en su perfil, el sistema muestra una interfaz para la entrada de los datos, el usuario cambia el valor de su contraseña y si los datos son correctos el sistema actualiza la contraseña.
Complejidad	Media
Prioridad	Alta
Precondiciones	

Anexos

Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción cambiar la contraseña en su perfil.	
2.		Muestra una interfaz para la entrada de los datos.
3.	Introduce los datos.	
4.		Comprueba que los datos sean correctos.
5.		Registra los cambios realizados por el usuario.
6.		Termina el CU.
Flujos alternos		
4. Los datos proporcionados por el usuario no son correctos.		
	Actor	Sistema
5		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Anexos

CU 27. Dejar de seguir documento.

Objetivo	Permitir que el usuario deje de seguir un documento.		
Actores	Usuario autenticado		
Resumen	El CU inicia cuando el usuario selecciona el documento que sigue, el sistema le muestra la información detallada del documento, el usuario selecciona la opción no seguir y el sistema registra los cambios.		
Complejidad	Media		
Prioridad	Media		
Precondiciones	El usuario debe ser seguidor del documento.		
Postcondiciones			
Flujo de eventos			
Flujo básico <Nombre del flujo básico>			
	Actor	Sistema	
1.	Selecciona el documento que sigue.		
2.		Muestra los detalles del documento.	
3.	Selecciona la opción dejar de seguir.		
4.		Registra la acción y lo elimina como seguidor del documento.	
5.		Termina el CU.	
Flujos alternos			
	Actor	Sistema	
5			
Relaciones	CU Incluidos	Mostrar datos de un documento	
	CU Extendidos		
Requisitos no			

Anexos

funcionales	
Asuntos pendientes	

CU 28. Dejar de seguir tema.

Objetivo	Permitir que el usuario deje de seguir un tema.	
Actores	Usuario autenticado	
Resumen	El CU inicia cuando el usuario selecciona en la opción dejar de seguir tema el tema que ya no desea seguir y el sistema lo elimina de la lista de seguidores del tema.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario debe ser seguidor del tema.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Accede a la opción dejar de seguir tema.	
2.		Muestra todos los temas que sigue el usuario.
3.	Selecciona el tema que desea dejar de seguir.	
4.		Registra la acción y lo elimina como seguidor del tema.
5.		Termina el CU.
Flujos alternos		
	Actor	Sistema

Anexos

5		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 29. Adicionar tema.

Objetivo	Permitir adicionar un tema.	
Actores	Administrador	
Resumen	El CU inicia cuando el administrador selecciona en la opción adicionar tema y el sistema lo agregan a la lista de temas.	
Complejidad	Baja	
Prioridad	Baja	
Precondiciones	El usuario debe estar autenticado y ser administrador del sistema.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Accede a la opción adicionar tema.	
2.		Muestra la interfaz para introducir el nuevo tema.
3.	Introduce el tema que desea adicionar.	
4.		El sistema adiciona el nuevo tema.
5.		Termina el CU.

Anexos

Flujos alternos		
	Actor	Sistema
5		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 30. Eliminar tema.

Objetivo	Permitir eliminar un tema.	
Actores	Administrador	
Resumen	El CU inicia cuando el administrador selecciona en la opción eliminar tema y el sistema lo elimina de la lista de temas.	
Complejidad	Baja	
Prioridad	Baja	
Precondiciones	El usuario debe estar autenticado y ser administrador del sistema.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Accede a la opción eliminar tema.	
2.		Muestra una interfaz con todos los temas.

Anexos

3.	Selecciona el tema que desea eliminar.	
4.		Elimina el tema.
5.		Termina el CU.
Flujos alternos		
	Actor	Sistema
5		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 31. Desbloquear usuario.

Objetivo	Desbloquear un usuario que se encuentre registrado en el sistema.
Actores	Super Administrador
Resumen	El CU se inicia cuando el super administrador del sistema requiere desbloquear un usuario, el mismo busca al usuario que desea modificar los permisos y accede a la opción desbloquear, el sistema desbloquea al usuario en el sistema.
Complejidad	Media
Prioridad	Media
Precondiciones	El usuario debe estar autenticado y ser super administrador del sistema.
Postcondiciones	
Flujo de eventos	

Anexos

Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona la opción gestionar usuarios en el panel de administración del sistema.	
2.		Muestra una interfaz donde se muestran todos los usuarios y que además permite realizar búsquedas de usuarios sobre la base de datos.
3.	Accede al usuario que desea desbloquear y accede a la opción desbloquear.	
4.		Desbloquea al usuario en el sistema.
5.		Finaliza el CU
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	Buscar usuario
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Anexos

Anexo 3

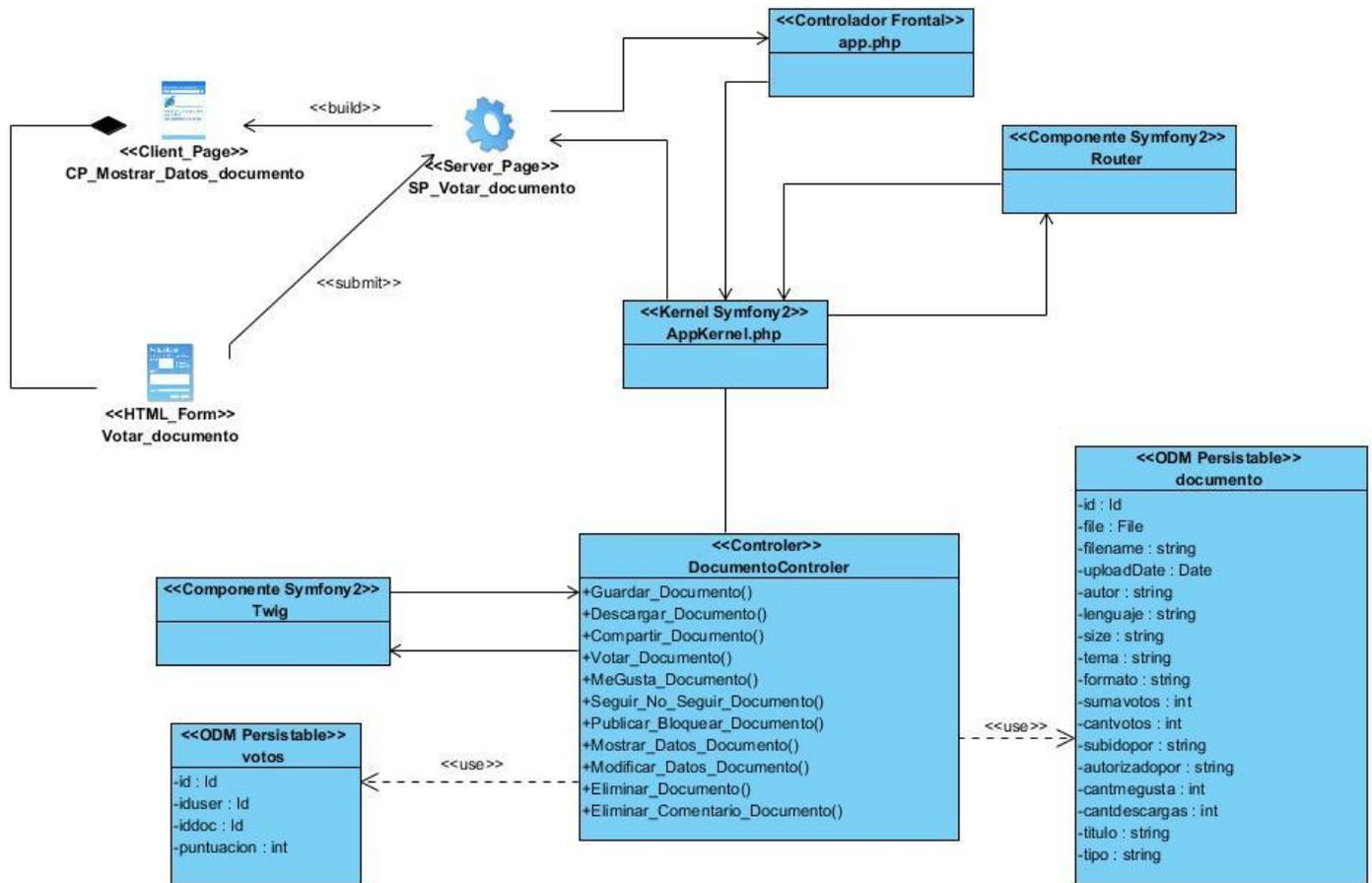


Diagrama de clases del diseño CU votar documento.

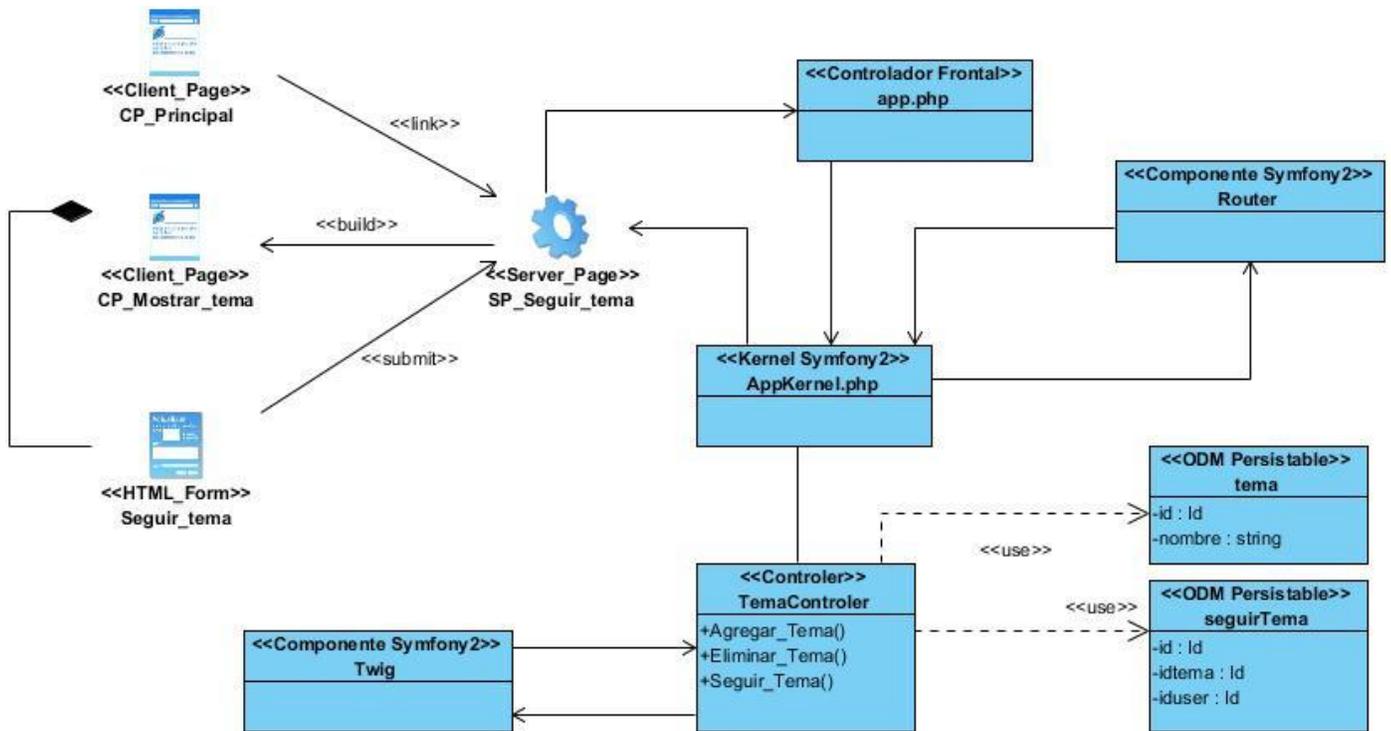


Diagrama de clases del diseño CU seguir tema.

Anexos

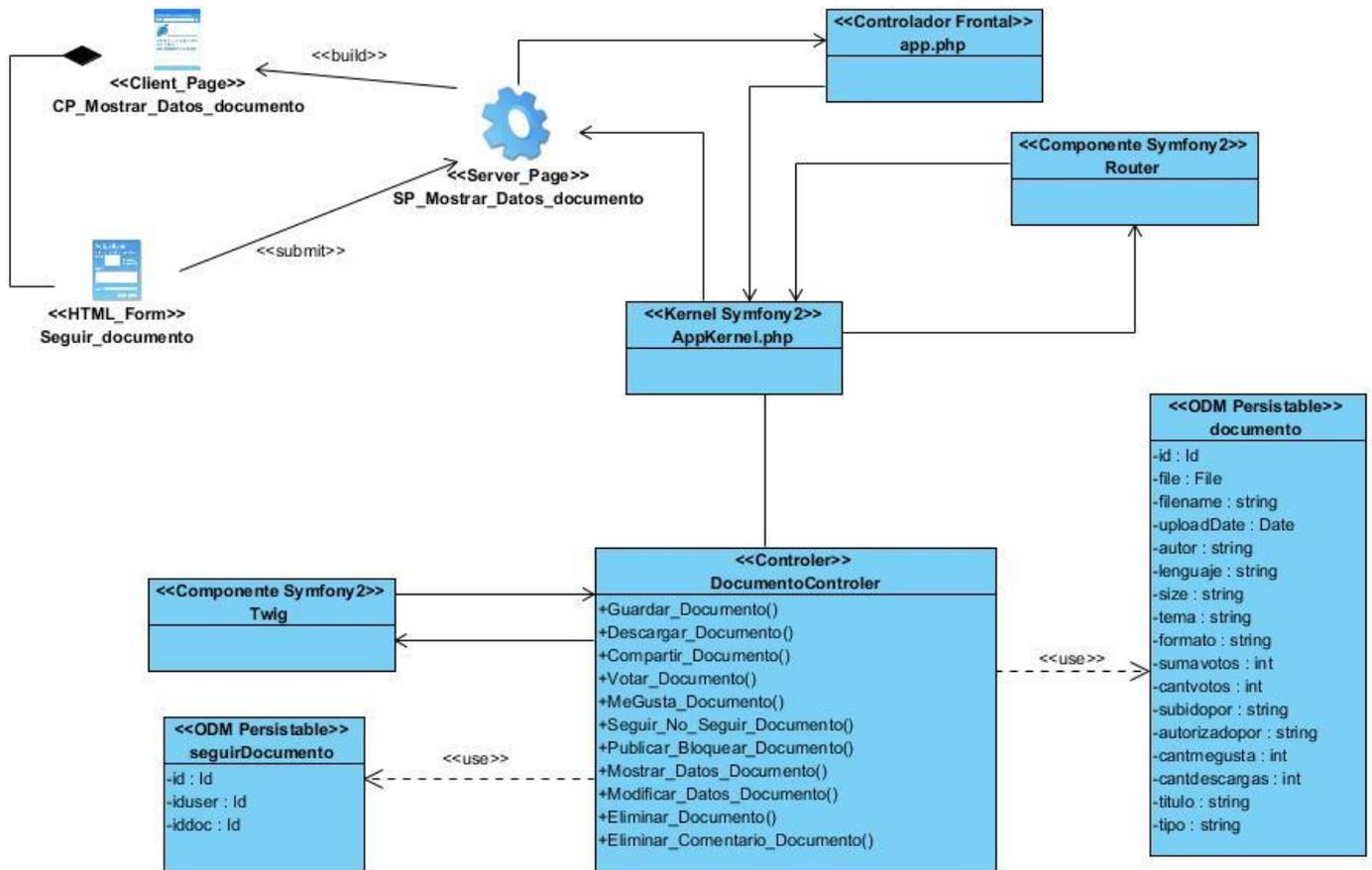


Diagrama de clases del diseño CU seguir documento.

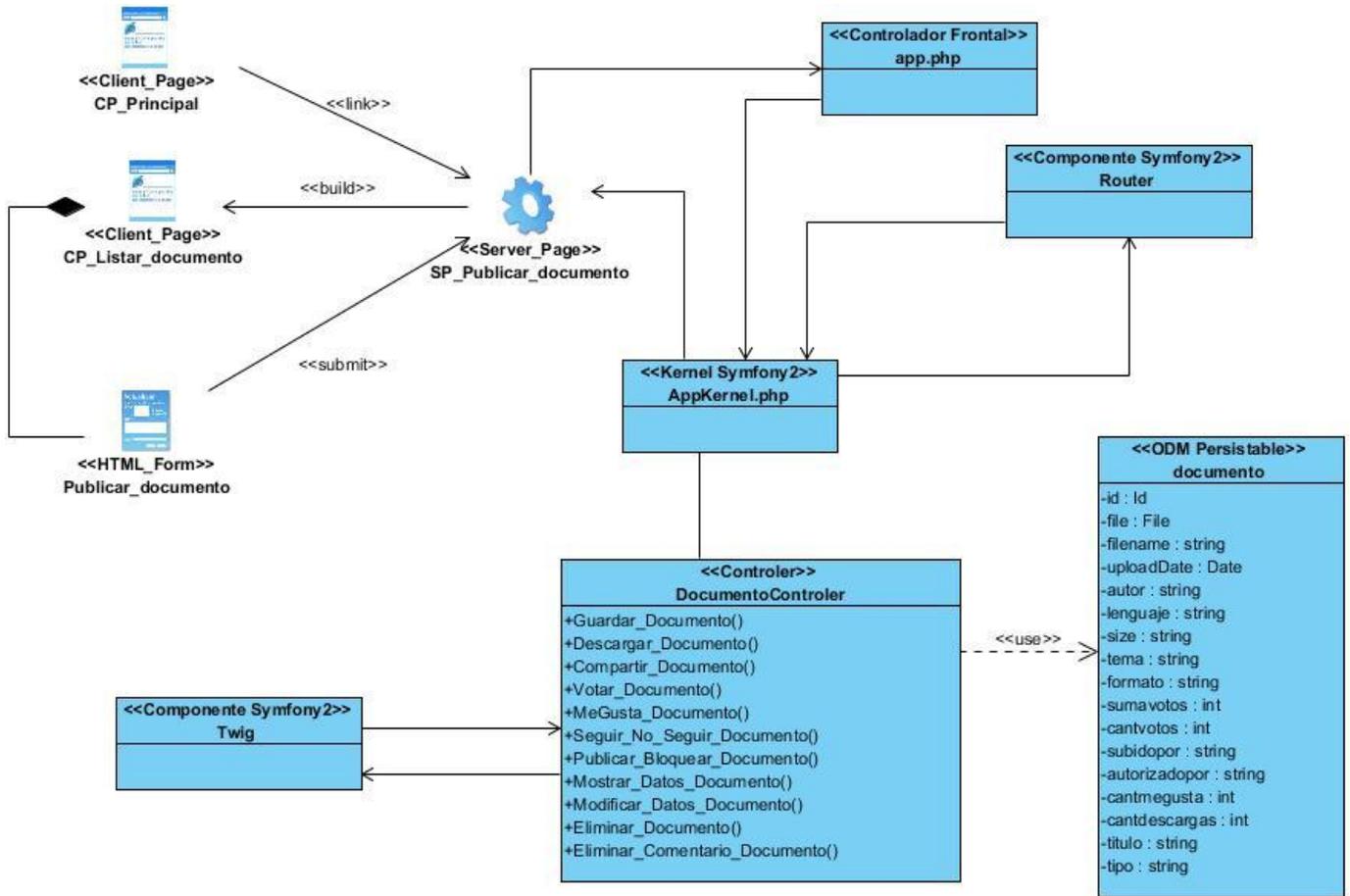


Diagrama de clases del diseño CU publicar documento.

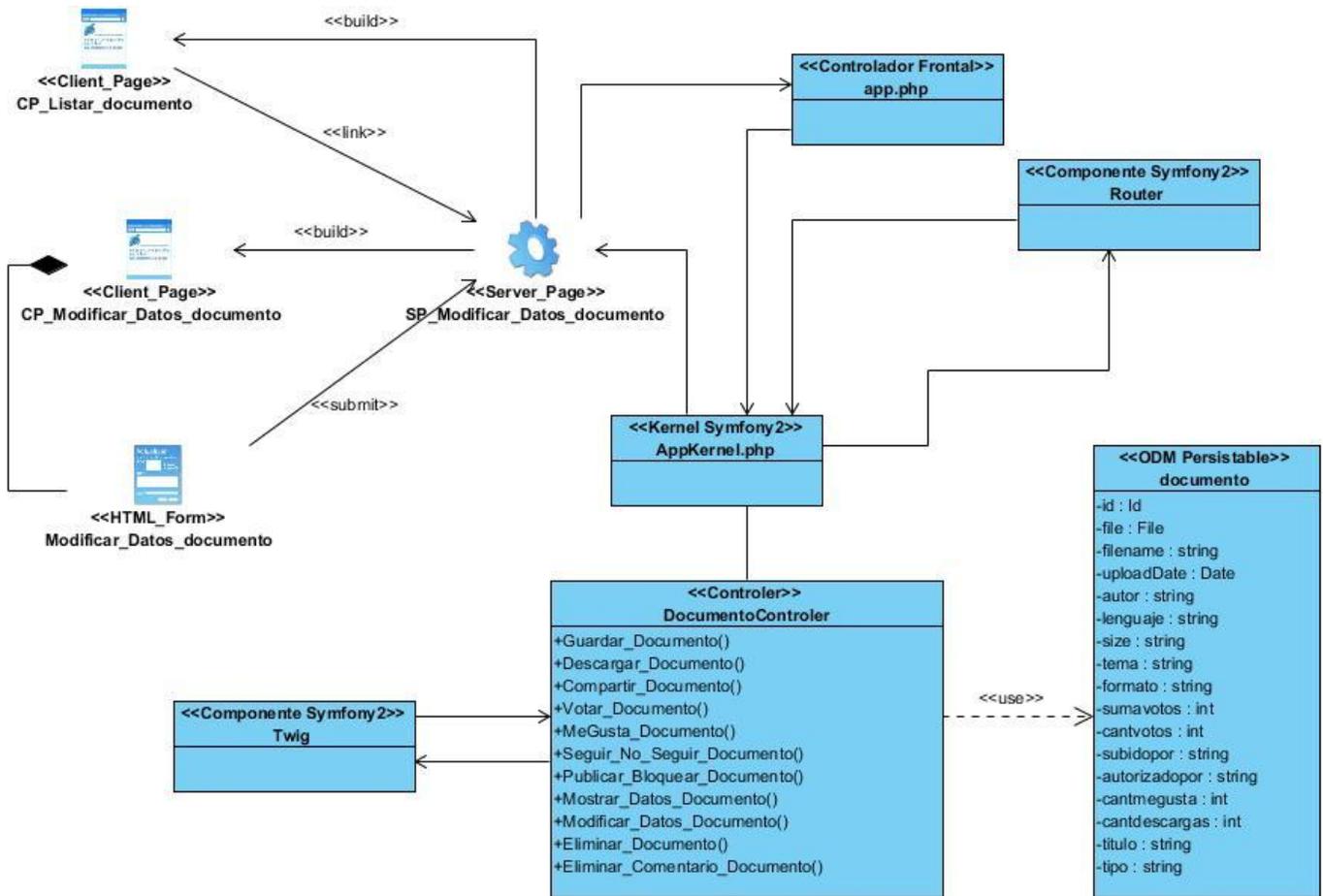


Diagrama de clases del diseño CU modificar datos documento.

Anexos

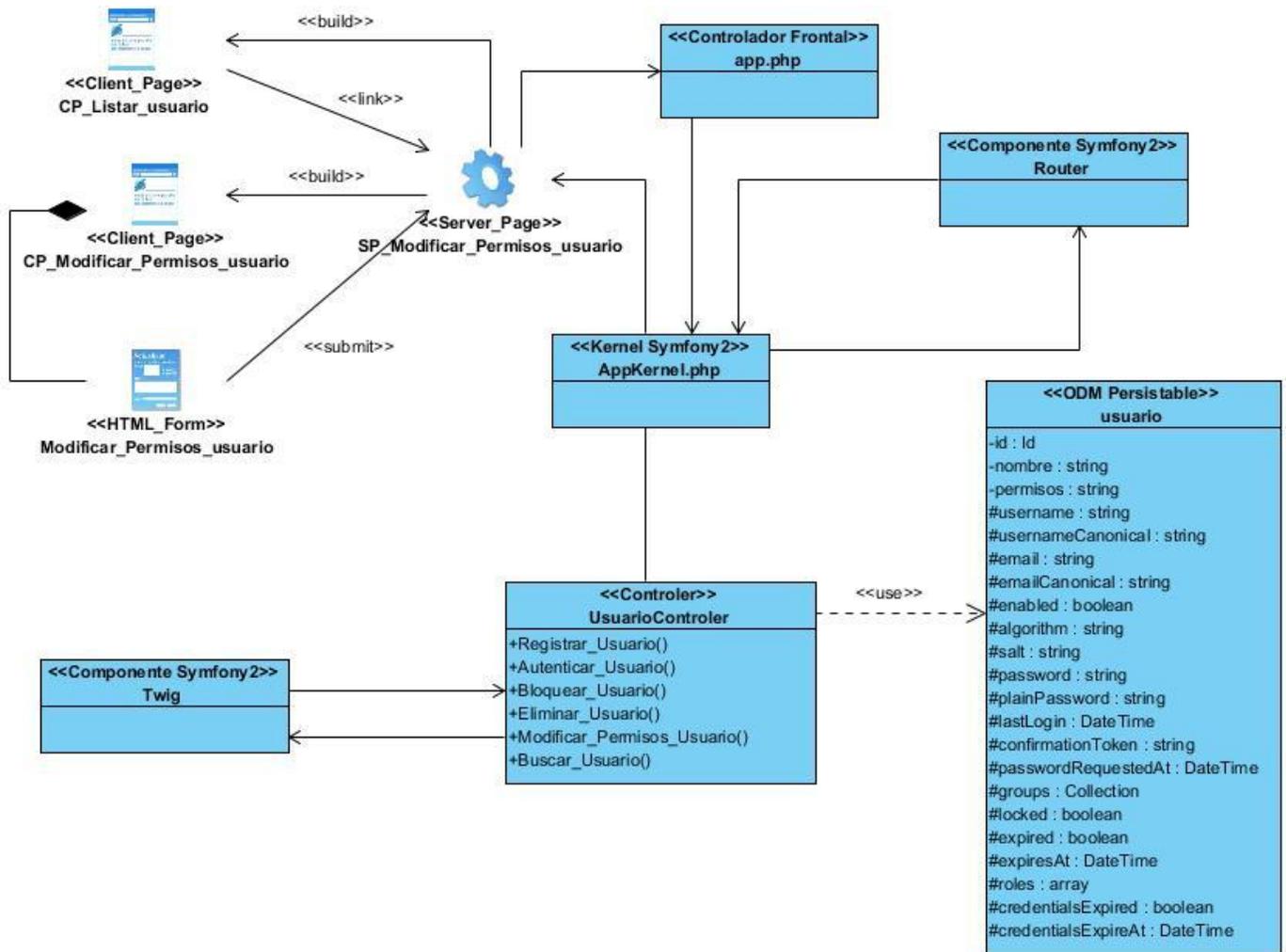


Diagrama de clases del diseño CU modificar permisos usuario.

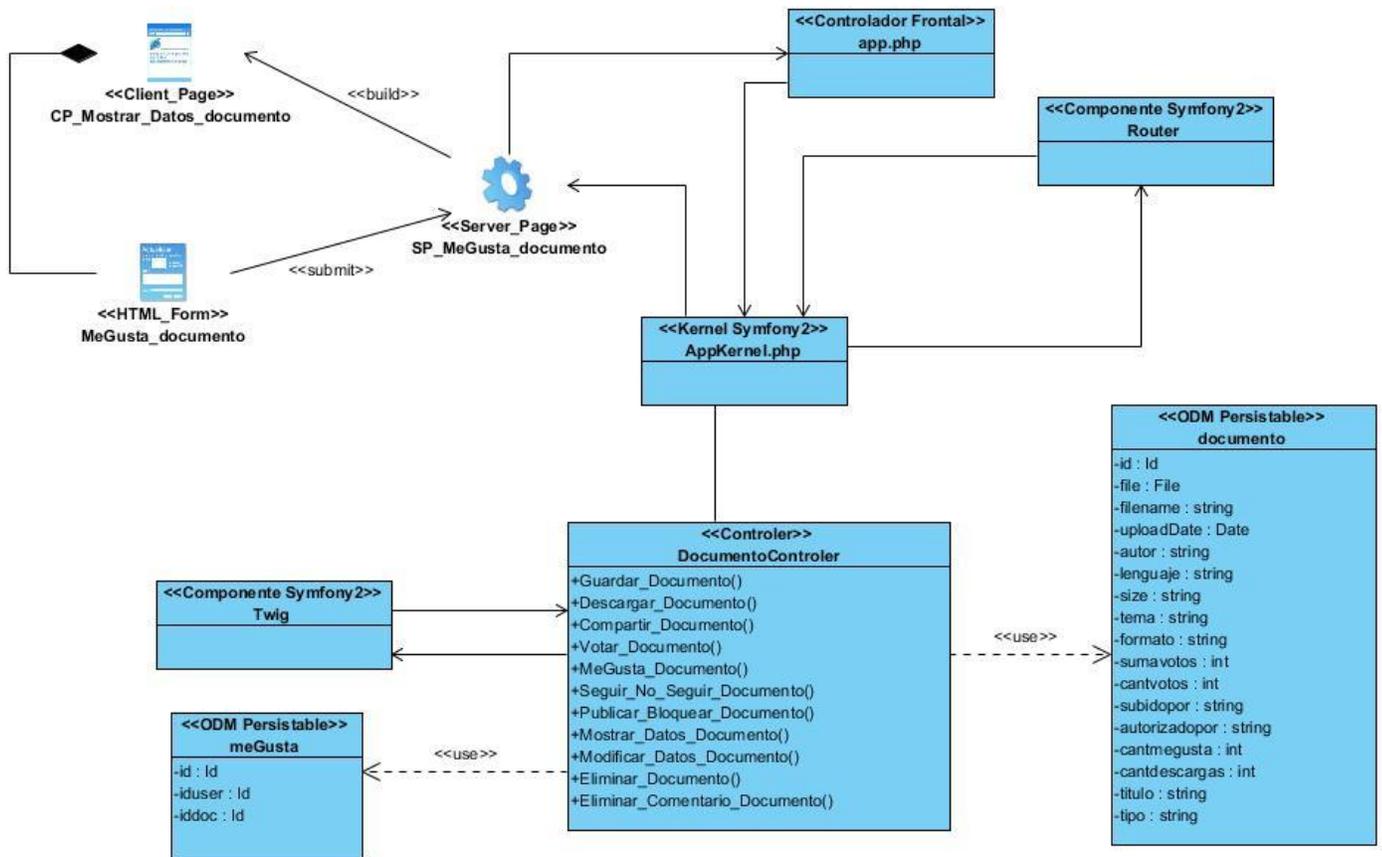


Diagrama de clases del diseño CU gustar documento.

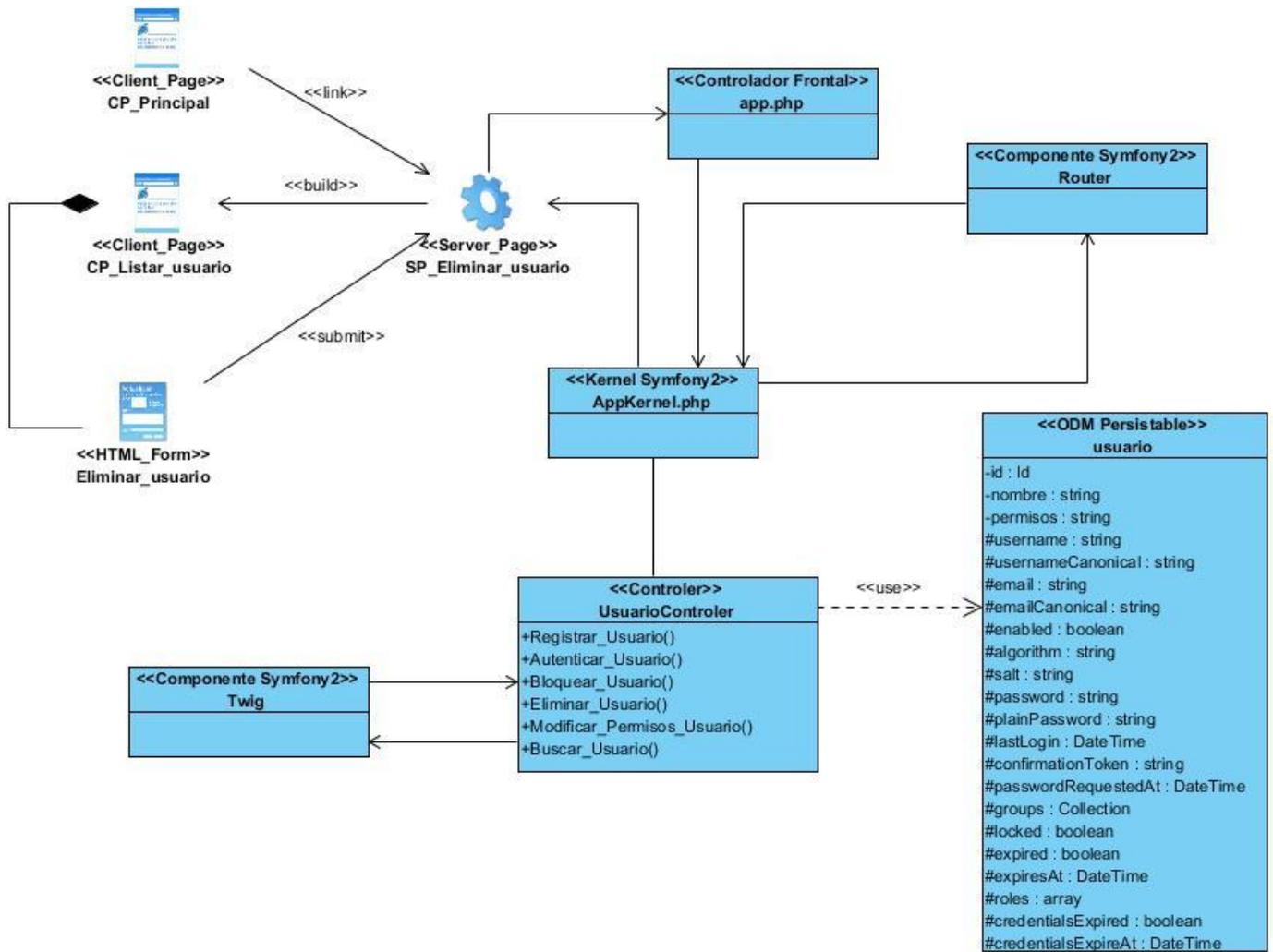


Diagrama de clases del diseño CU eliminar usuario.

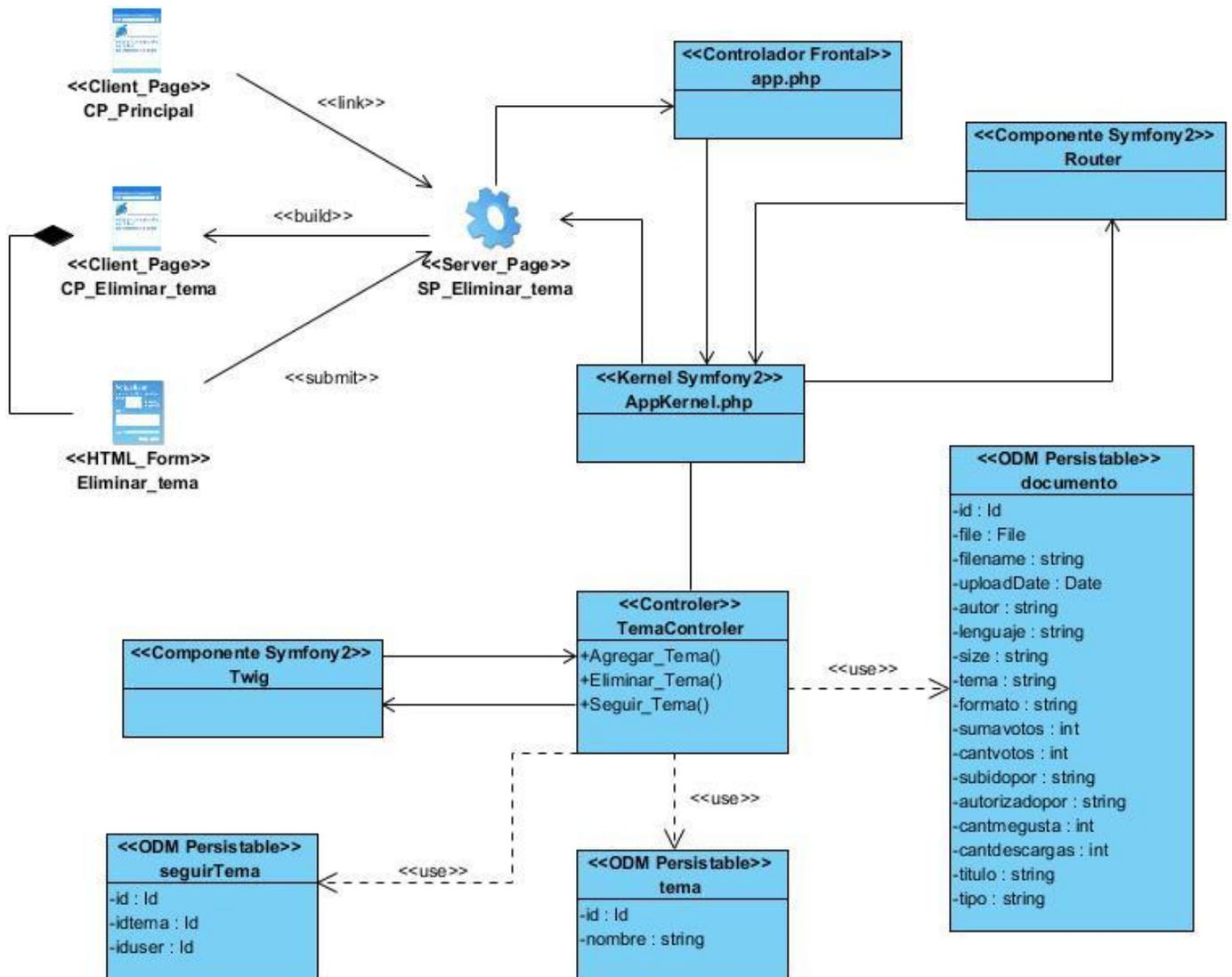


Diagrama de clases del diseño CU eliminar tema.

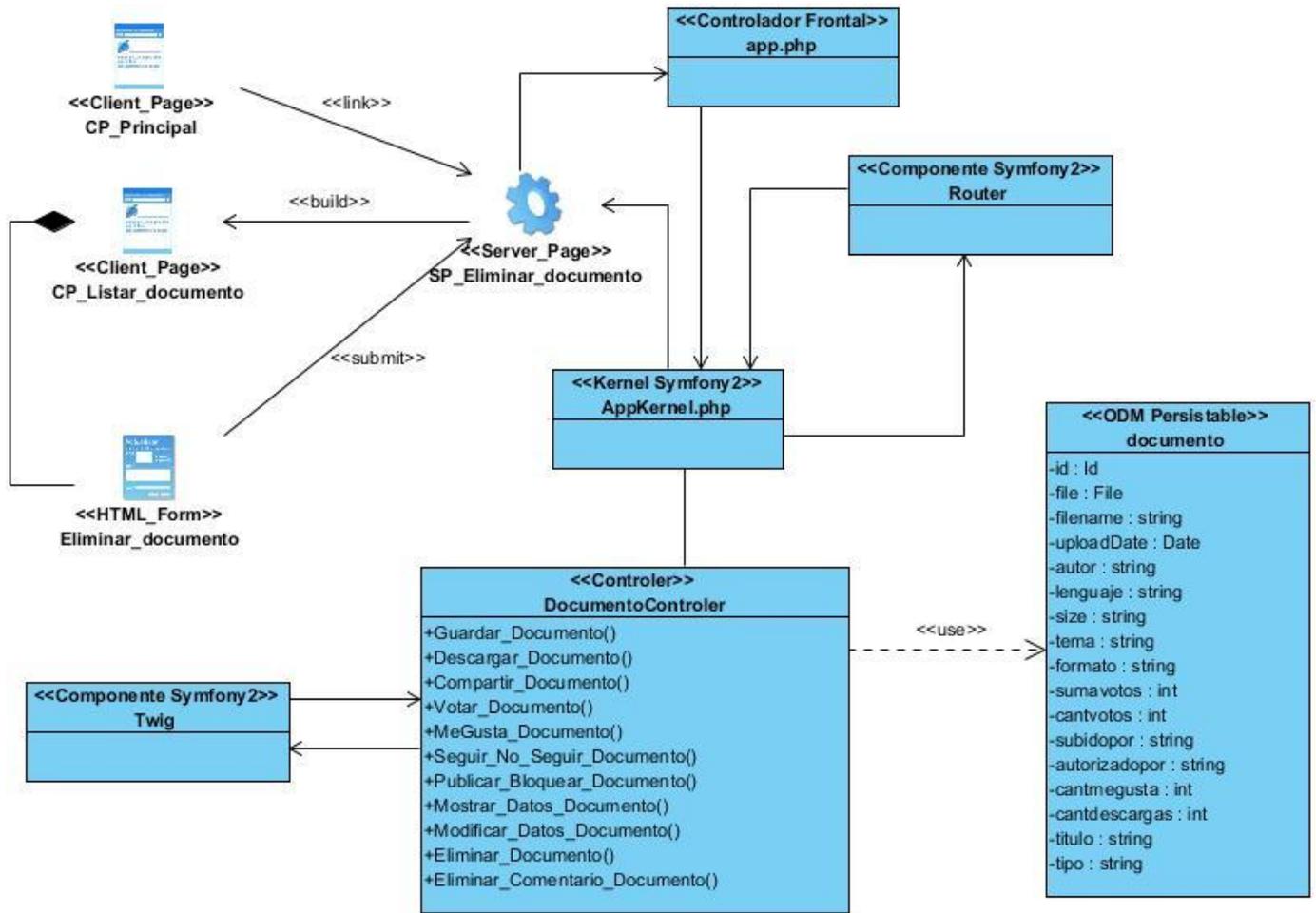


Diagrama de clases del diseño CU eliminar documento.

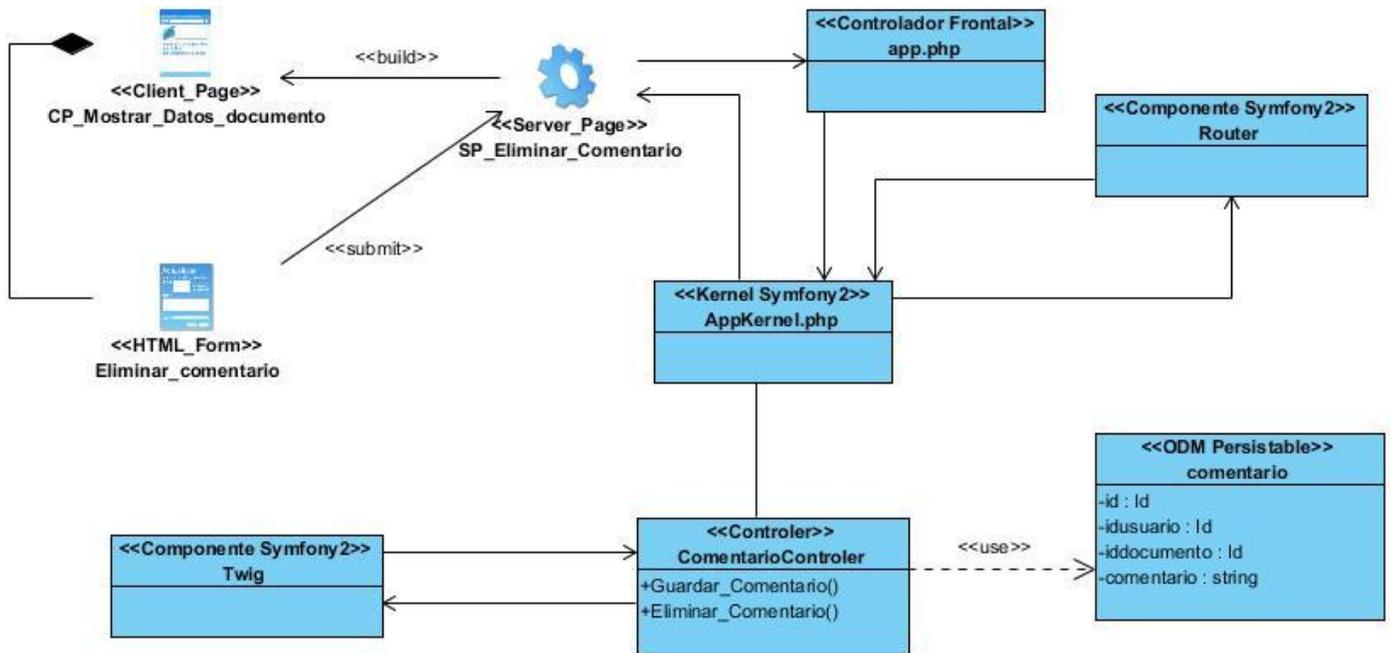


Diagrama de clases del diseño CU eliminar comentario.

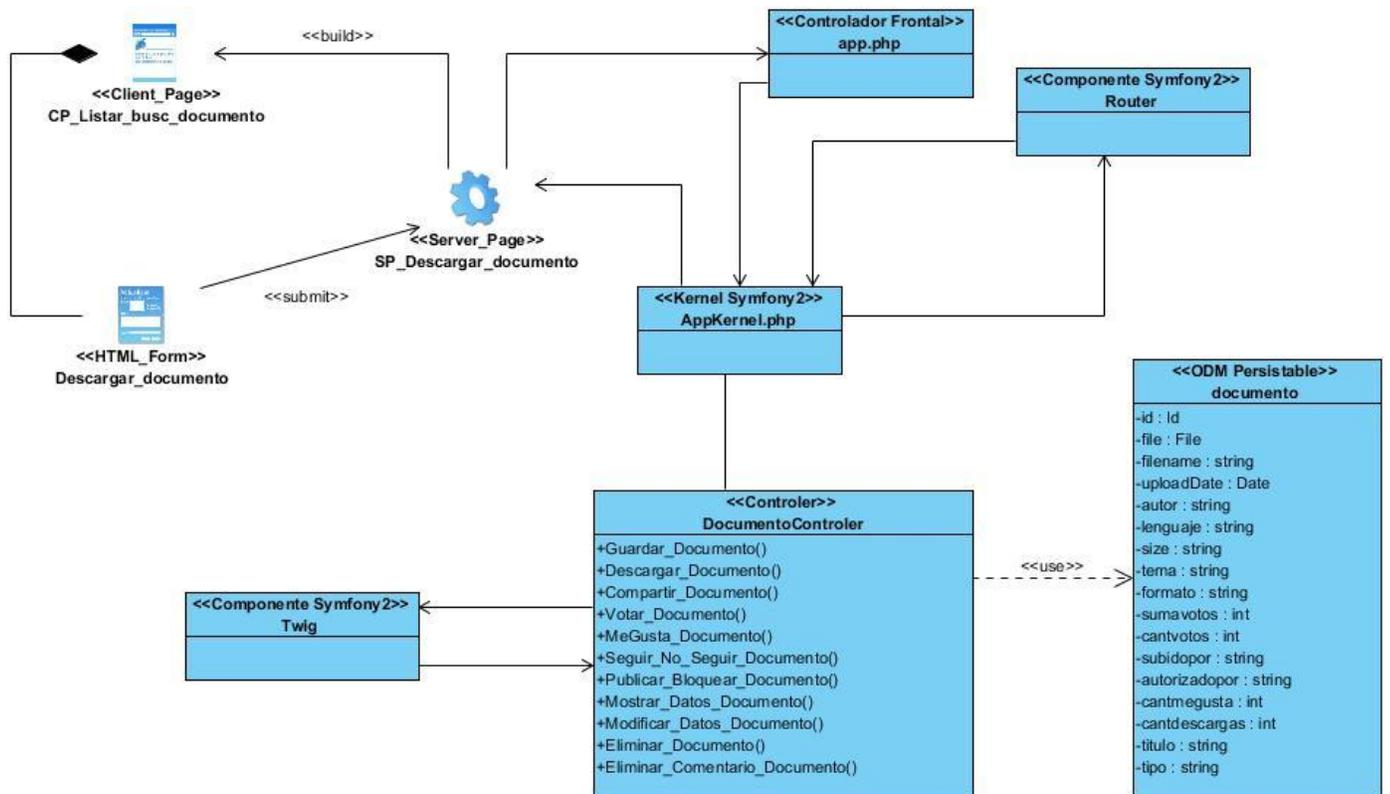


Diagrama de clases del diseño CU descargar documento.

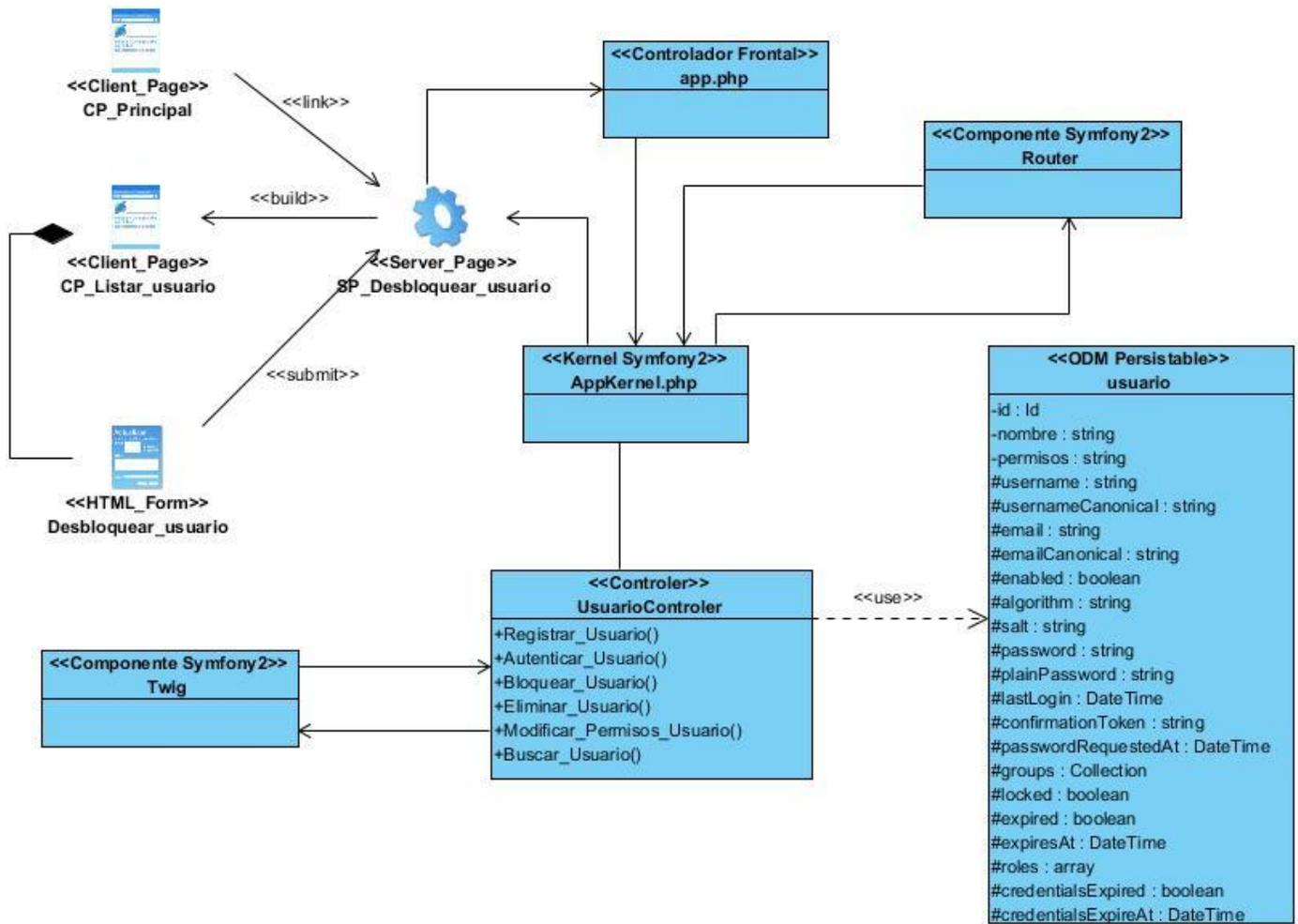


Diagrama de clases del diseño CU desbloquear usuario.

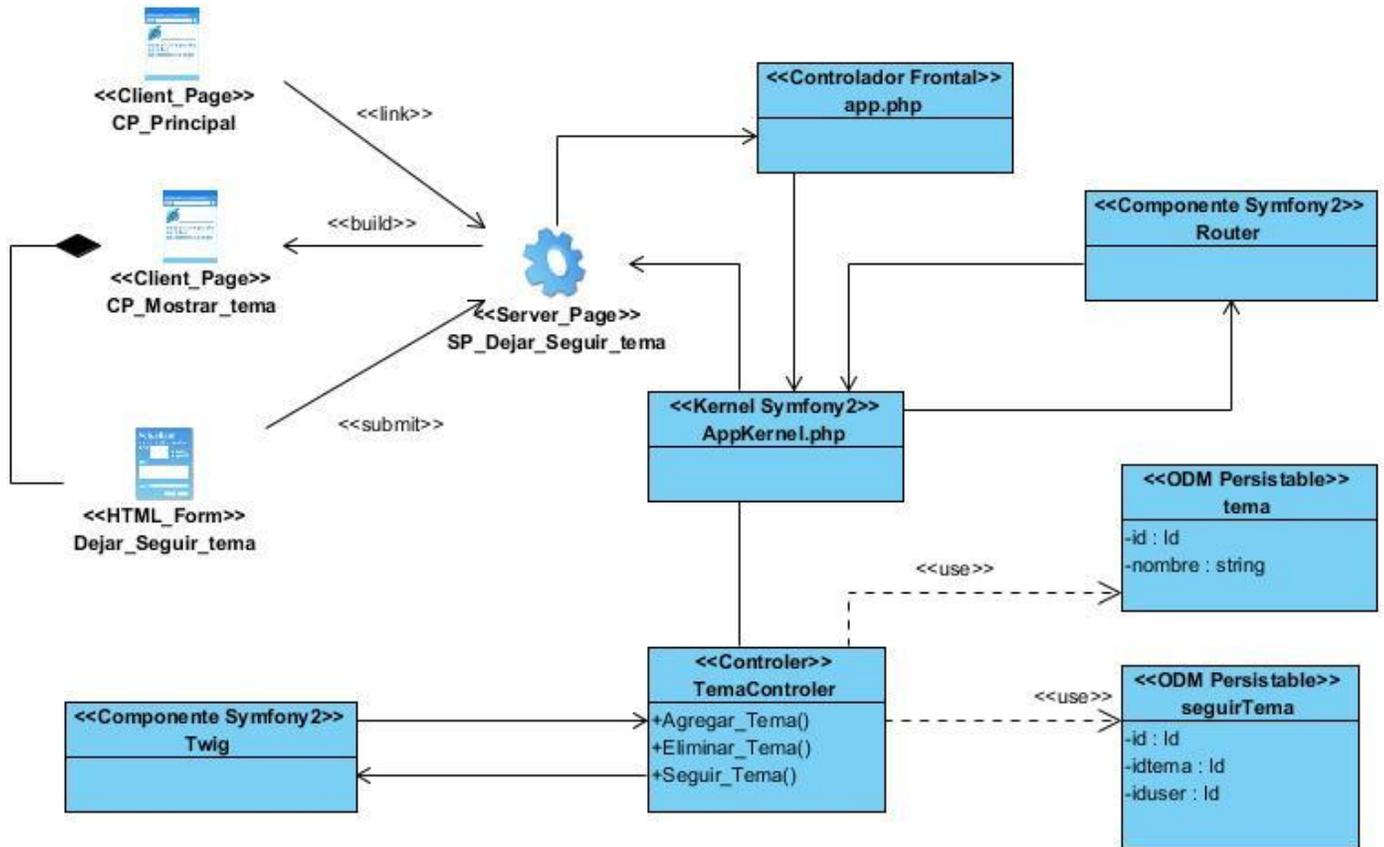


Diagrama de clases del diseño CU dejar seguir tema.

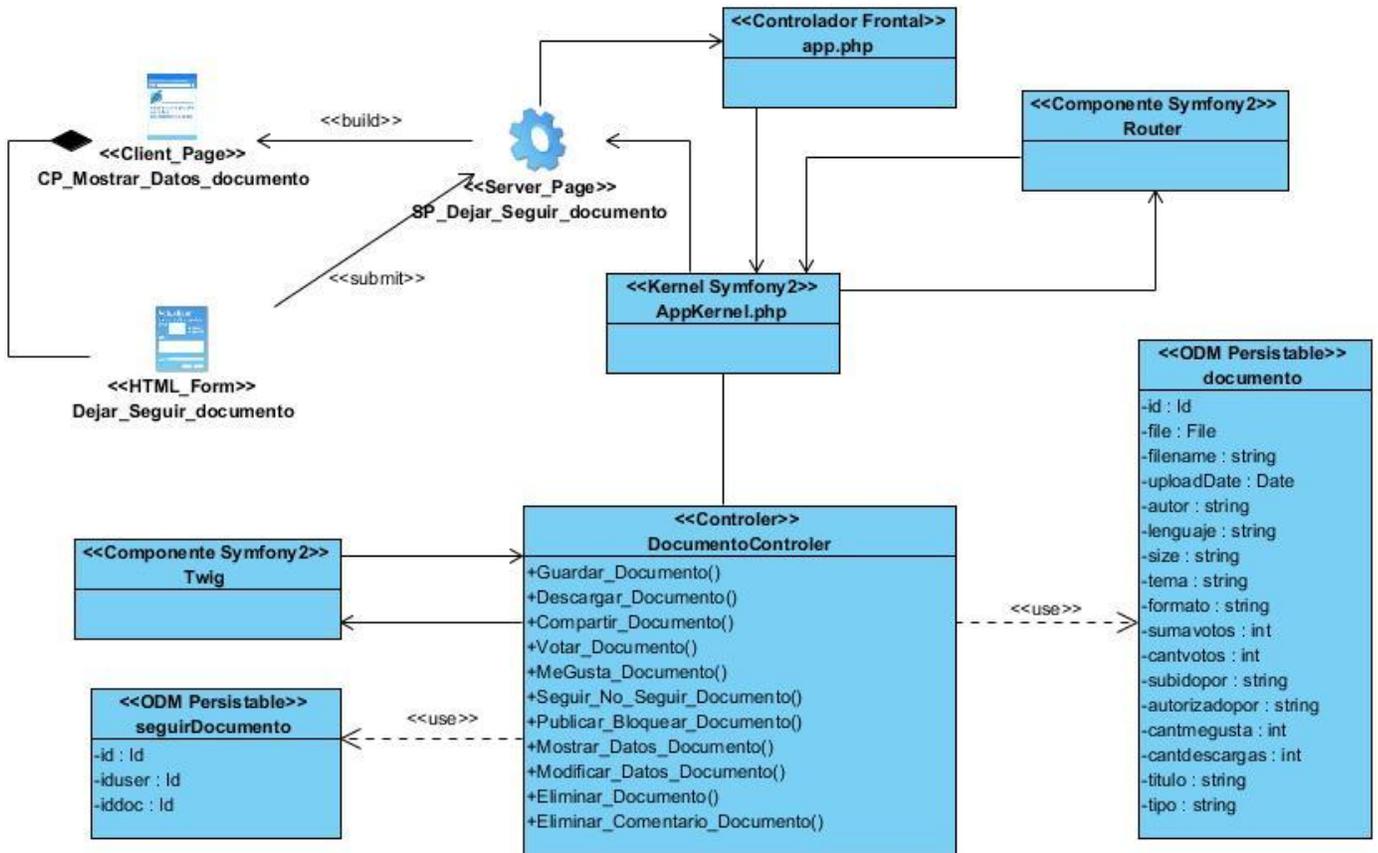


Diagrama de clases del diseño CU dejar seguir documento.

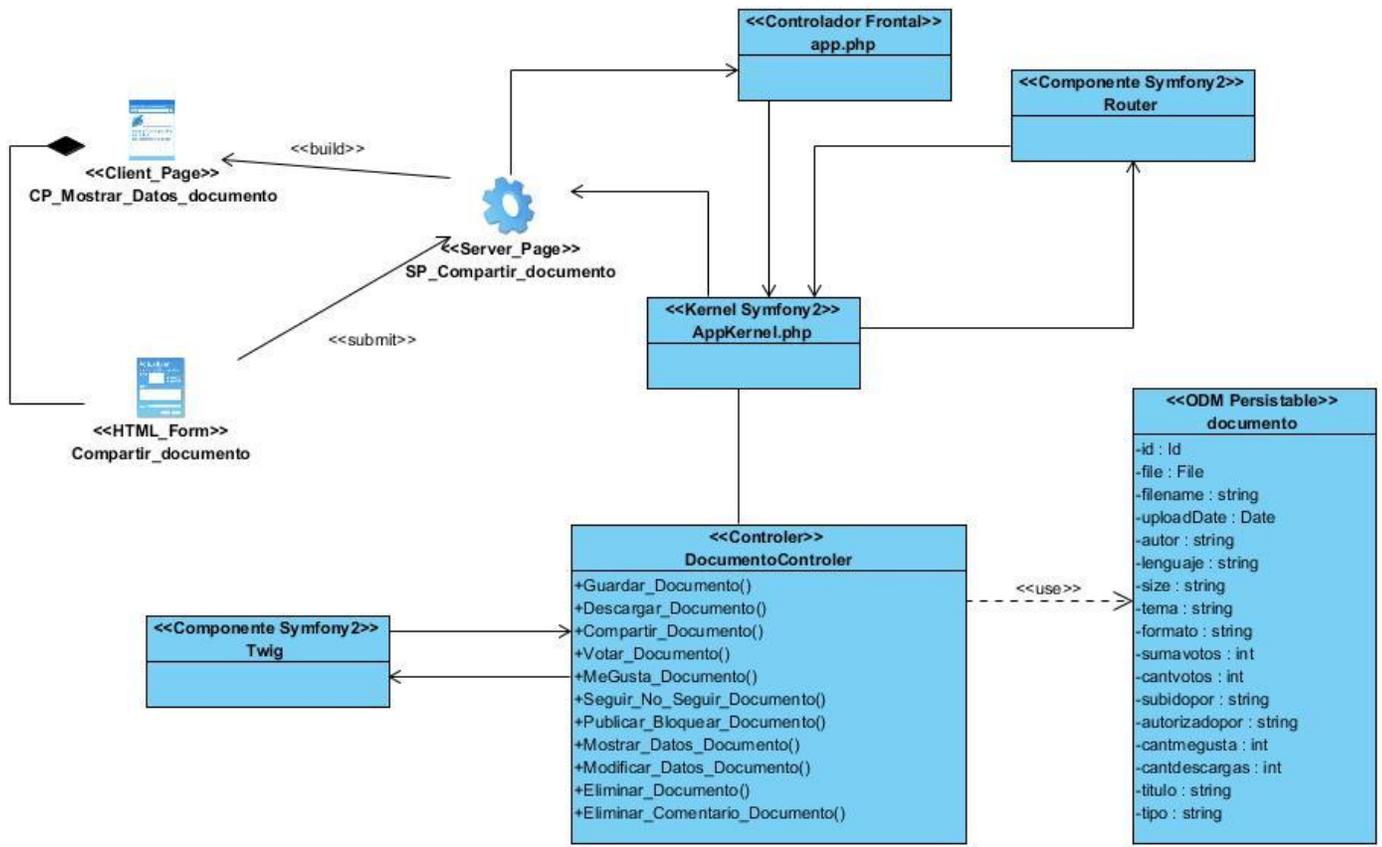


Diagrama de clases del diseño CU compartir documento.

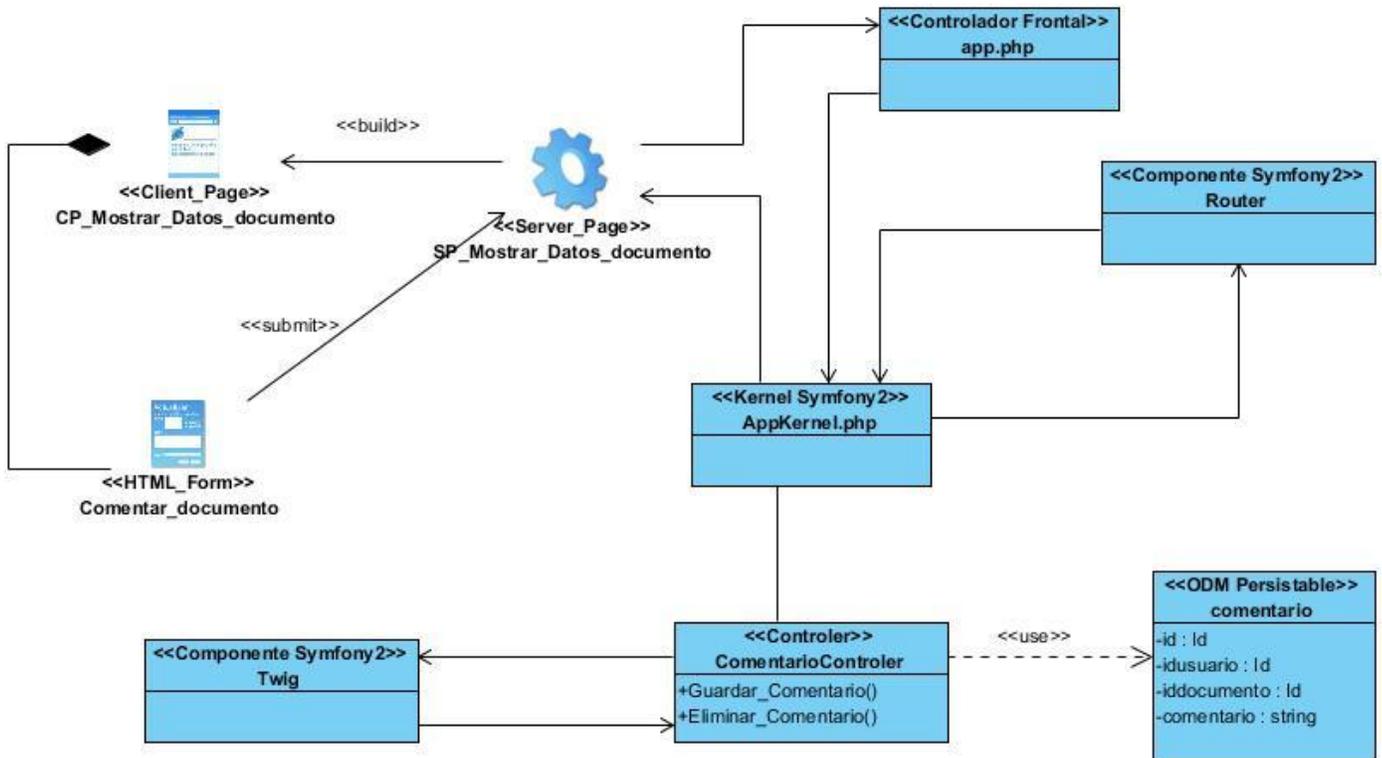


Diagrama de clases del diseño CU comentar documento.

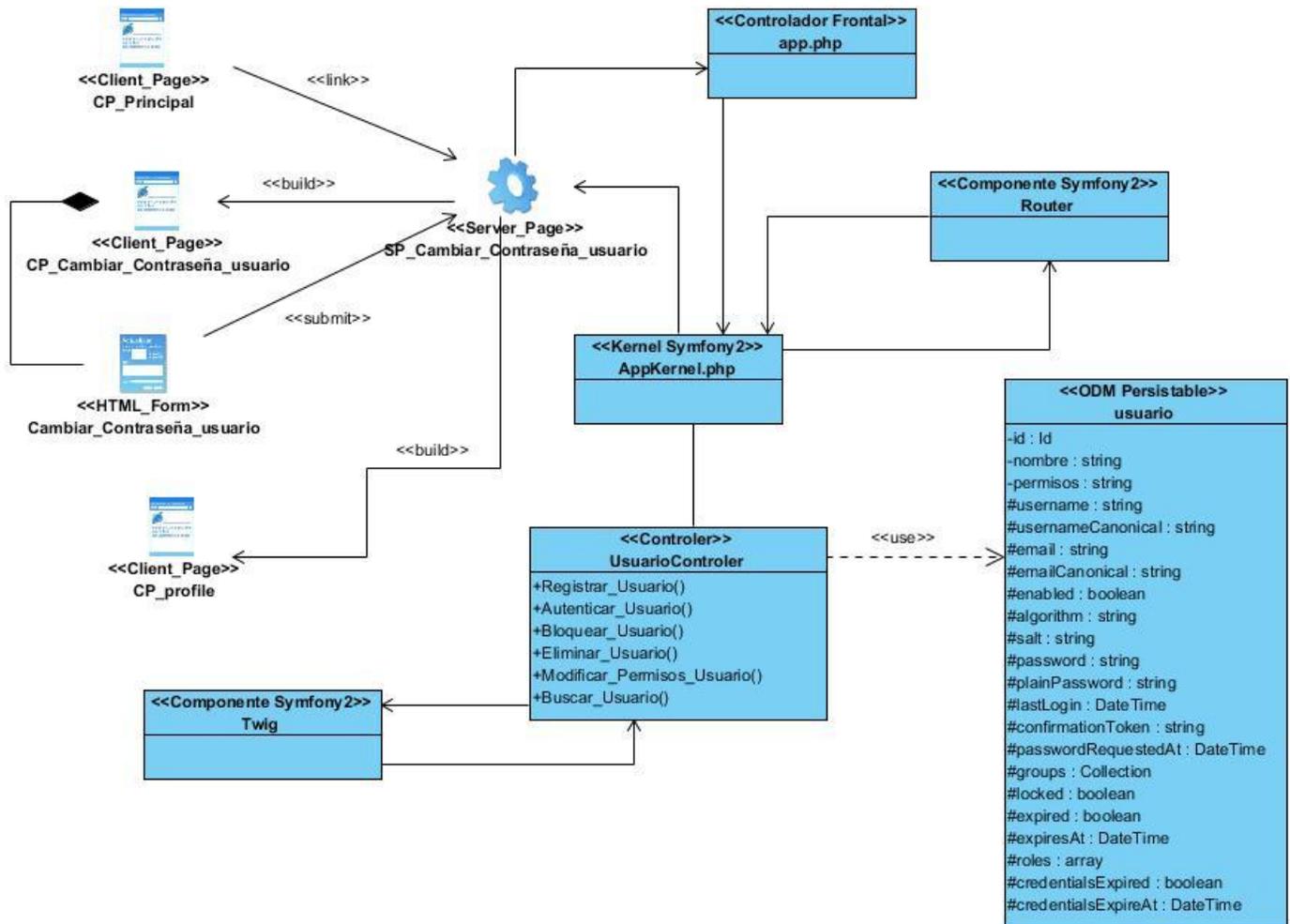


Diagrama de clases del diseño CU cambiar contraseña.

Anexos

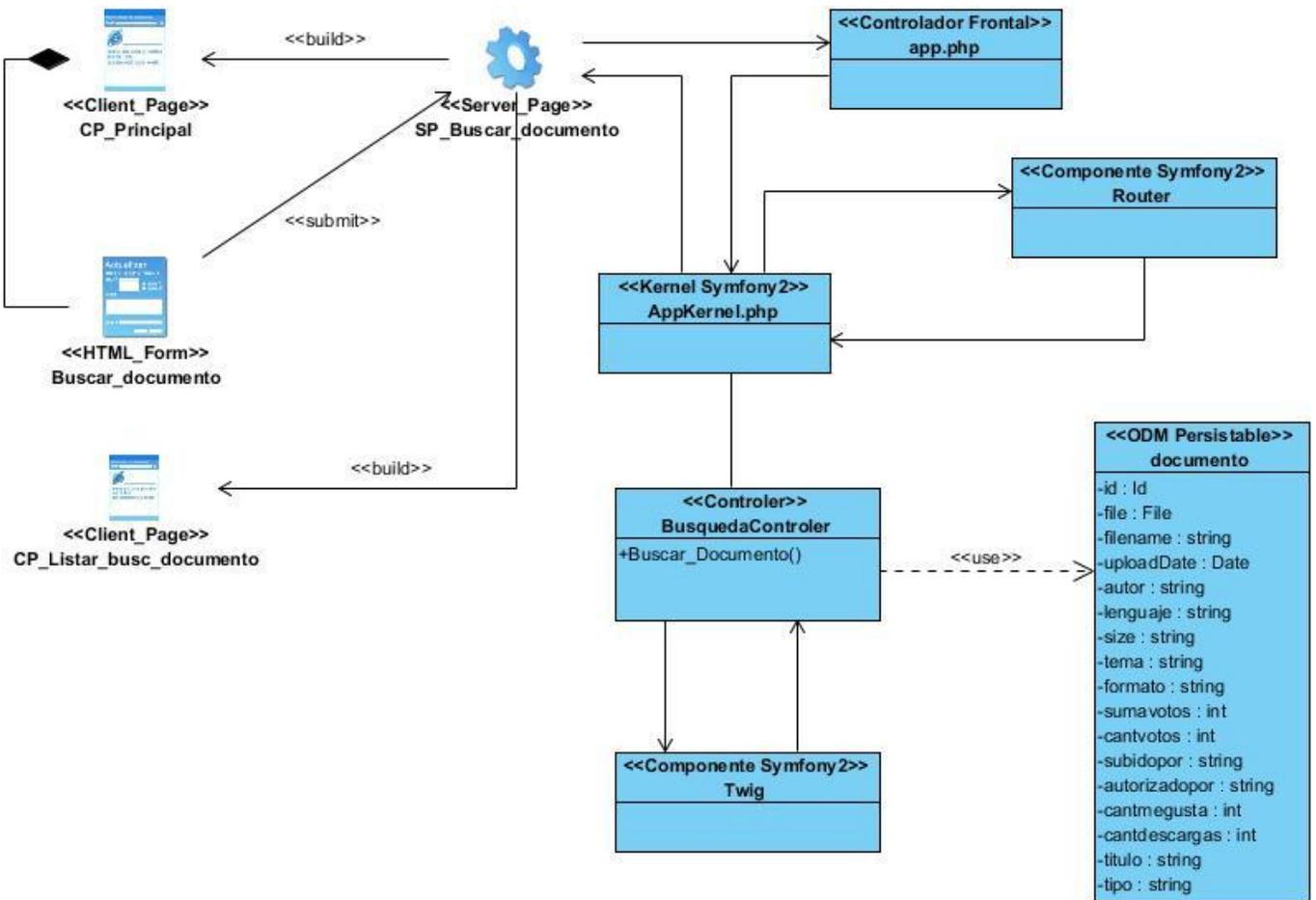


Diagrama de clases del diseño CU buscar documento.

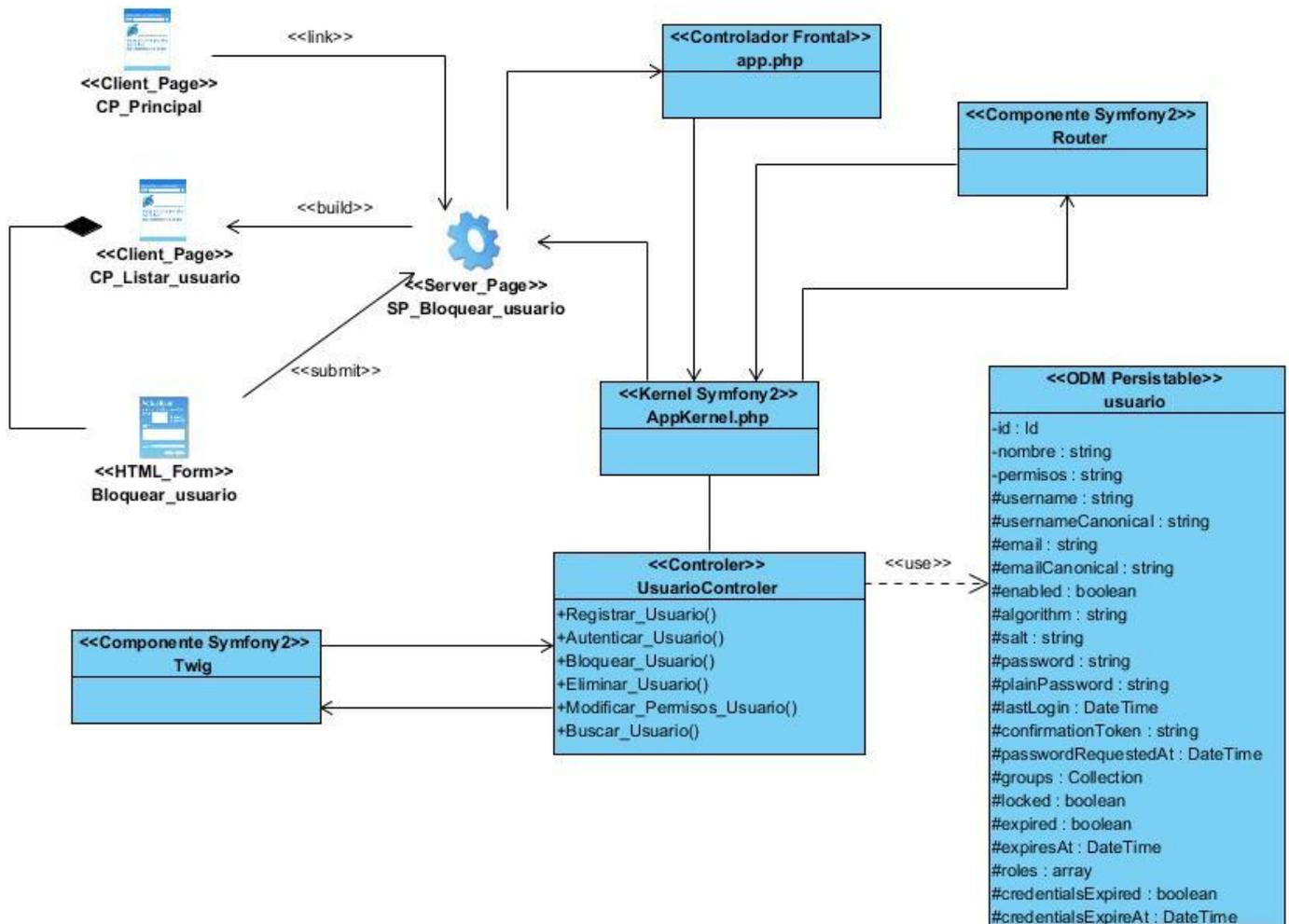


Diagrama de clases del diseño CU bloquear usuario.

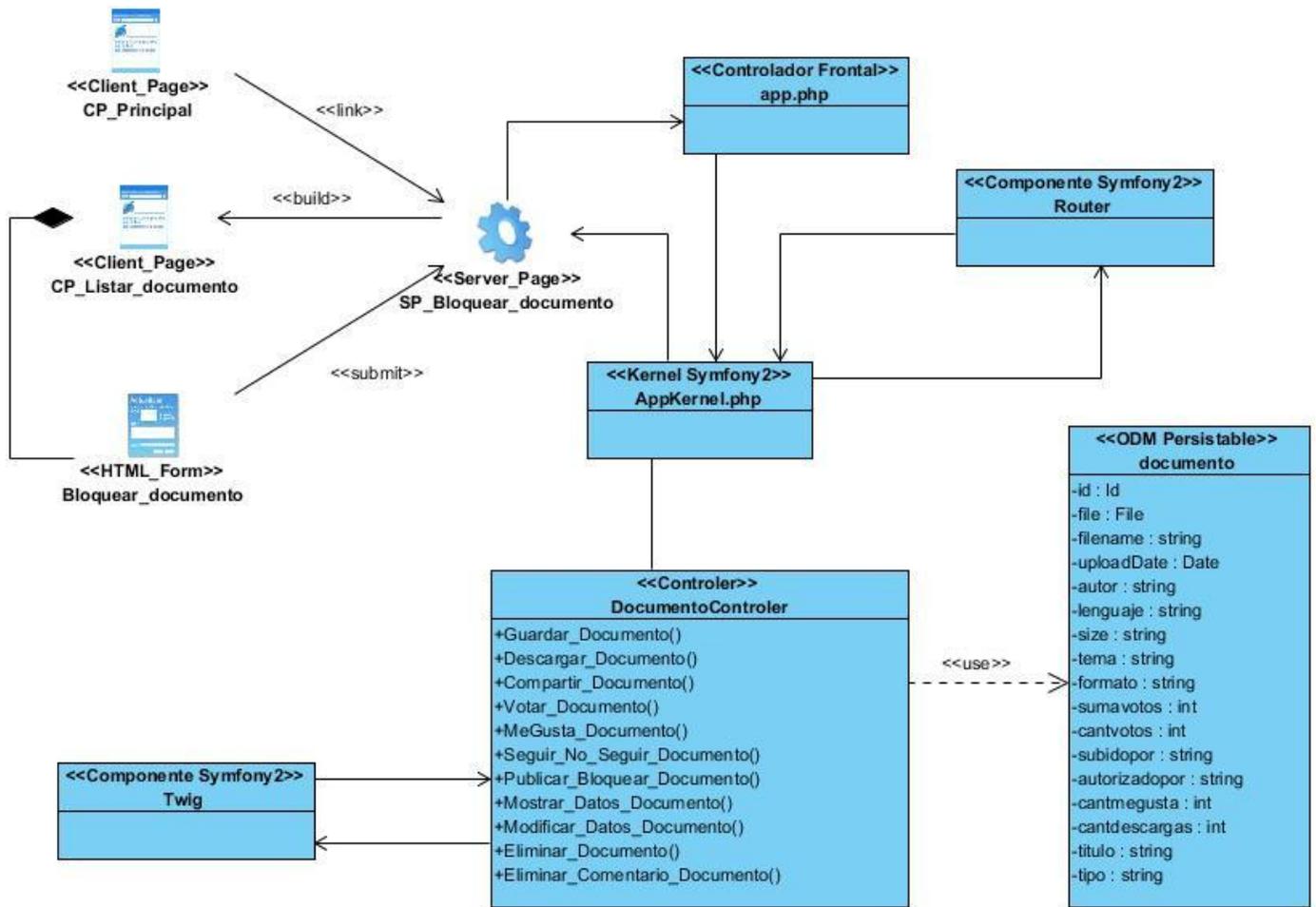


Diagrama de clases del diseño CU bloquear documento.

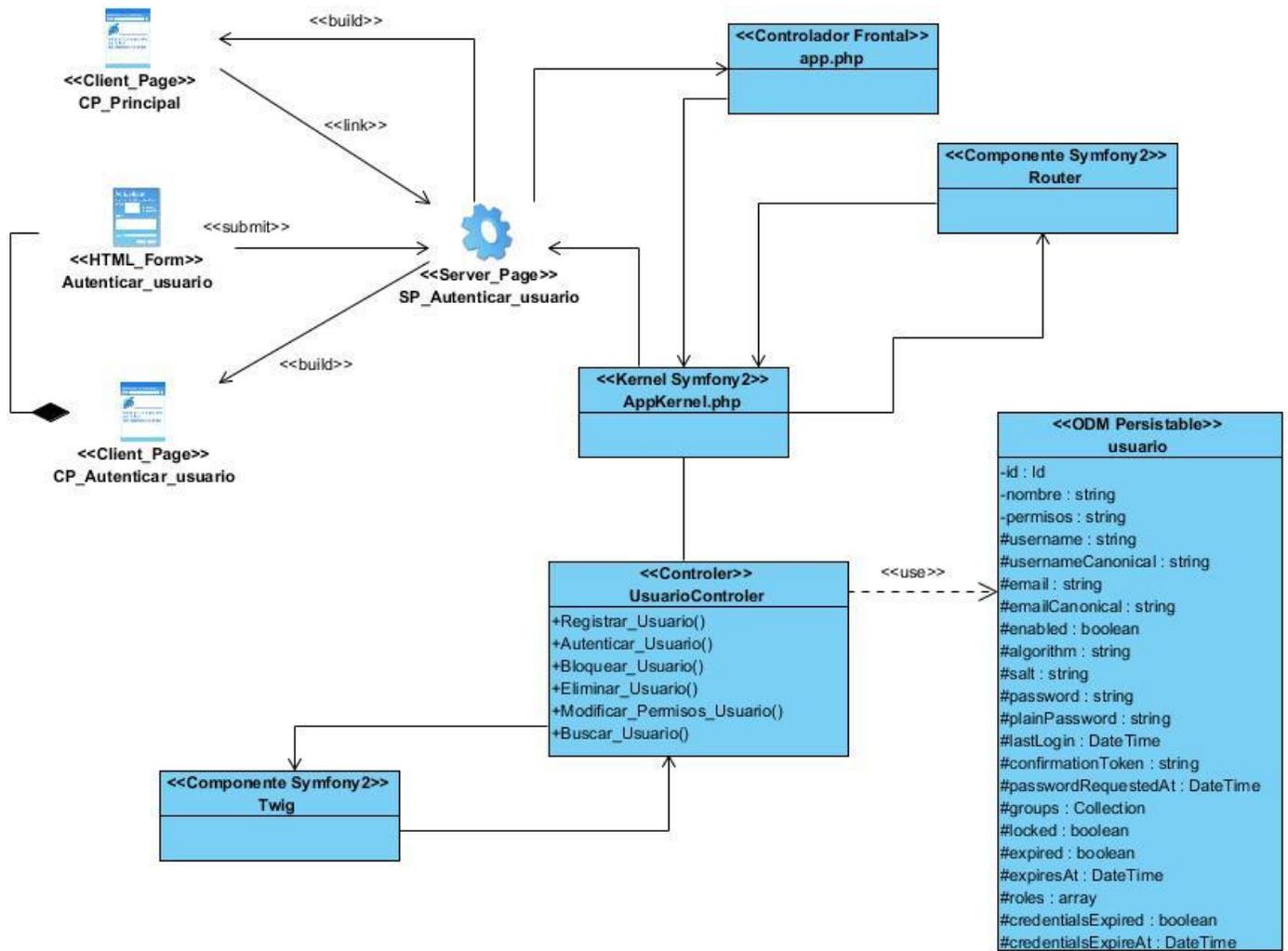


Diagrama de clases del diseño CU autenticar usuario.

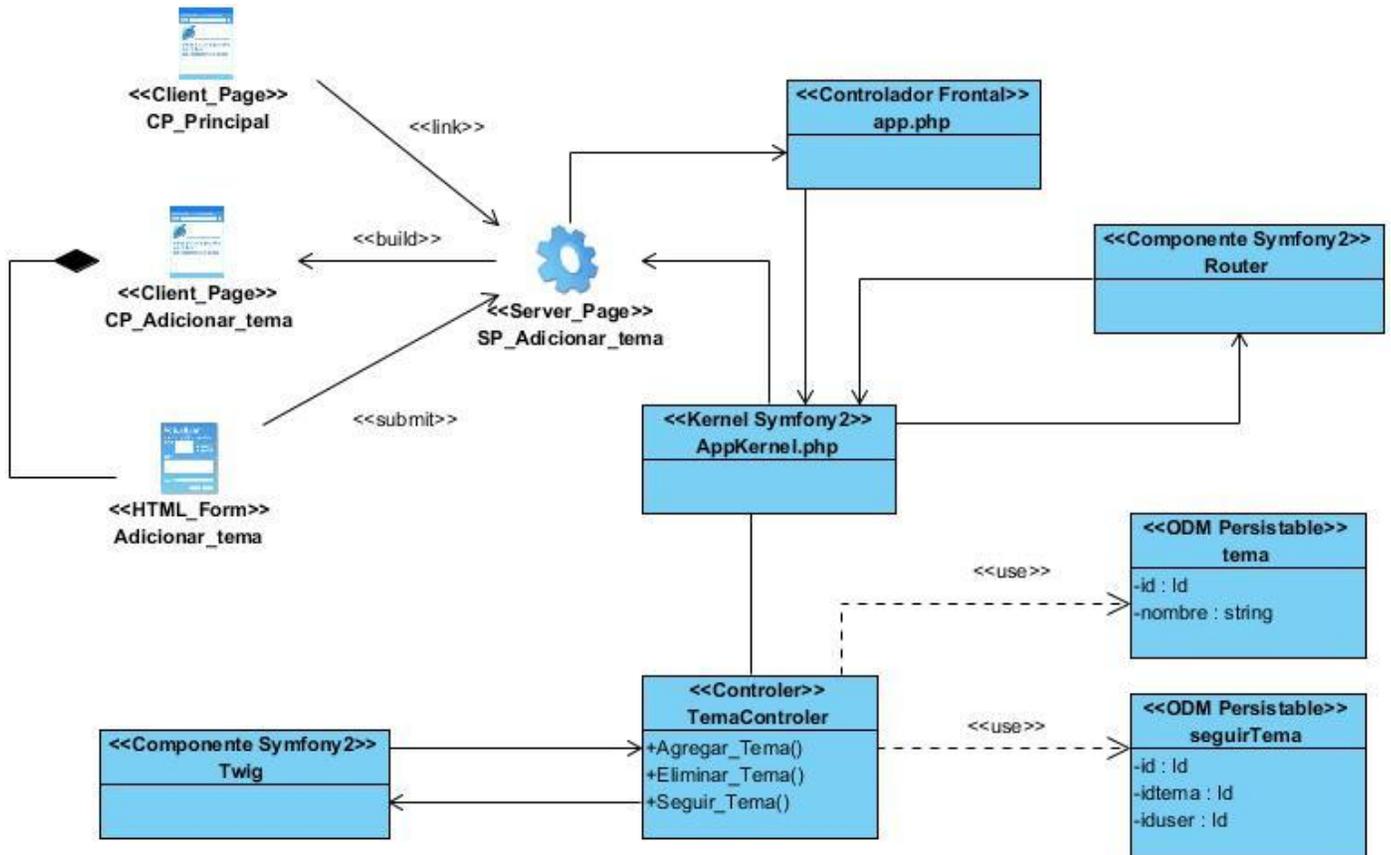


Diagrama de clases del diseño CU adicionar tema.

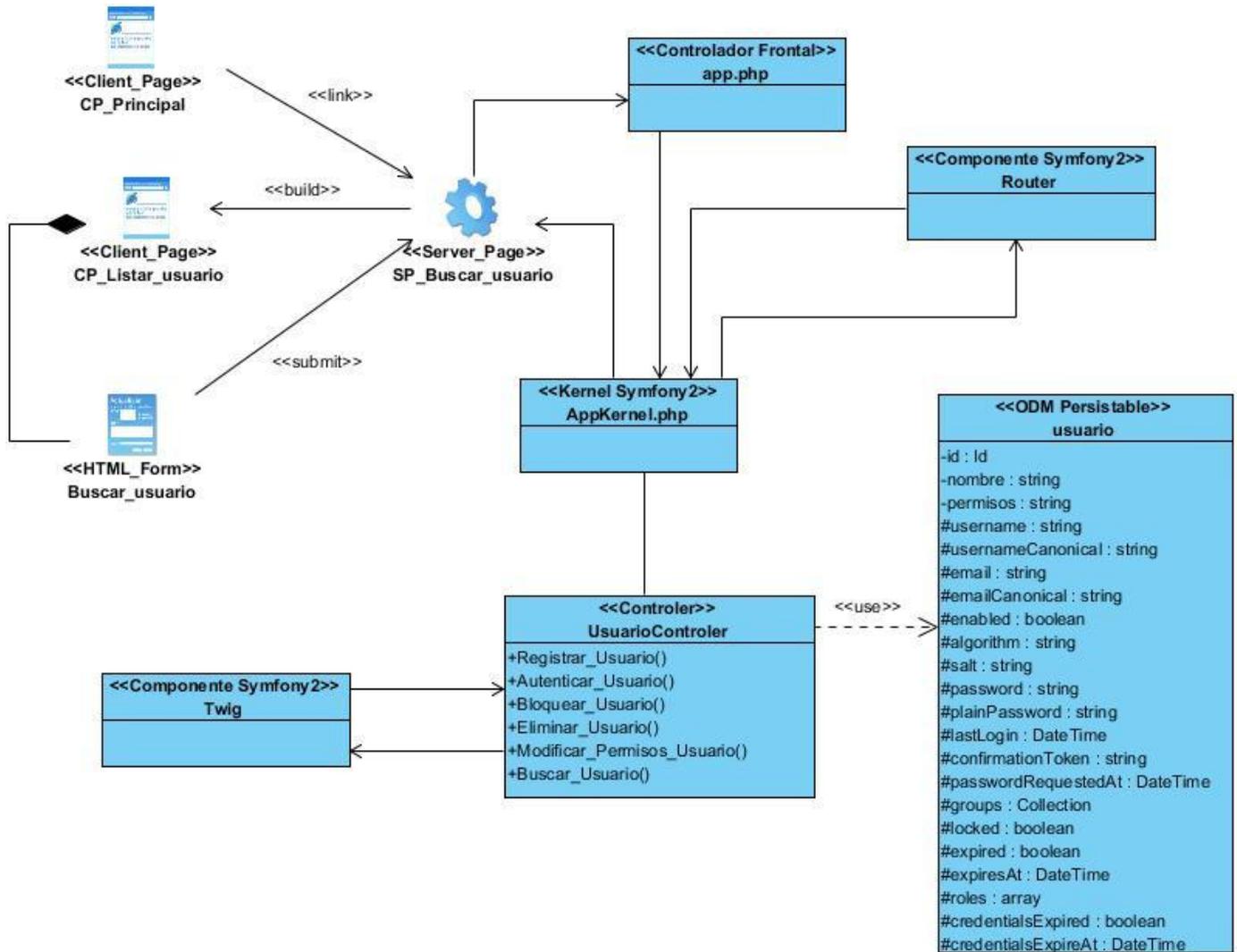


Diagrama de clases del diseño CU buscar usuario.

Anexos

Anexo 4

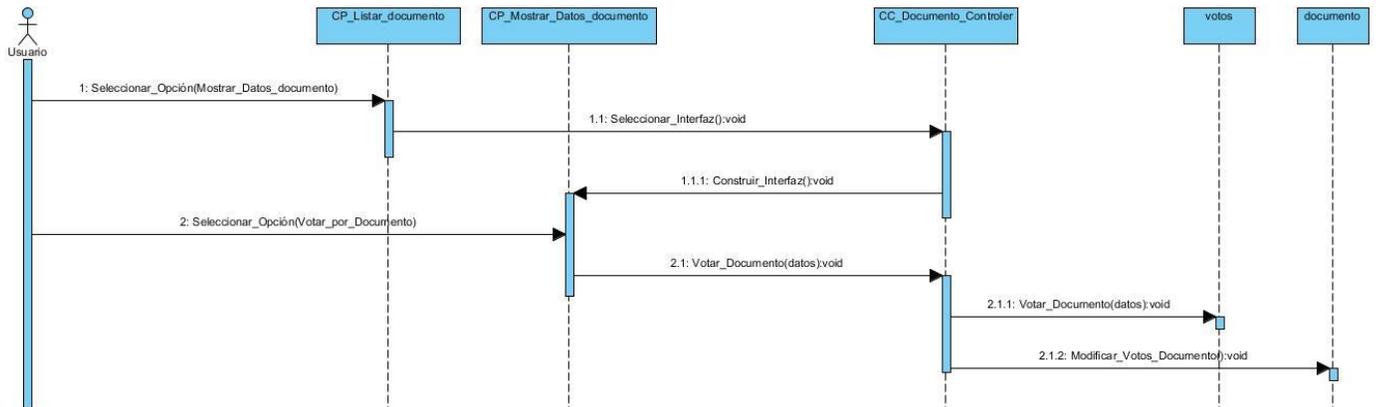


Diagrama de secuencia CU votar documento.

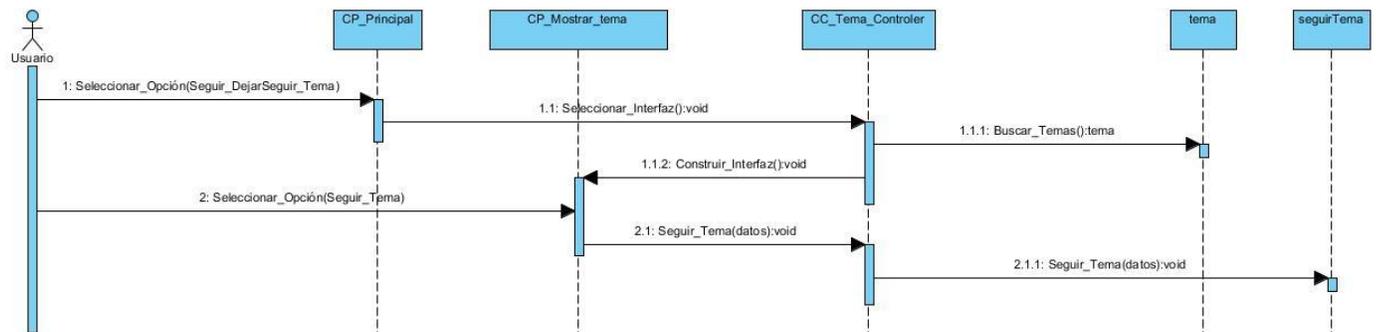


Diagrama de secuencia CU seguir tema.

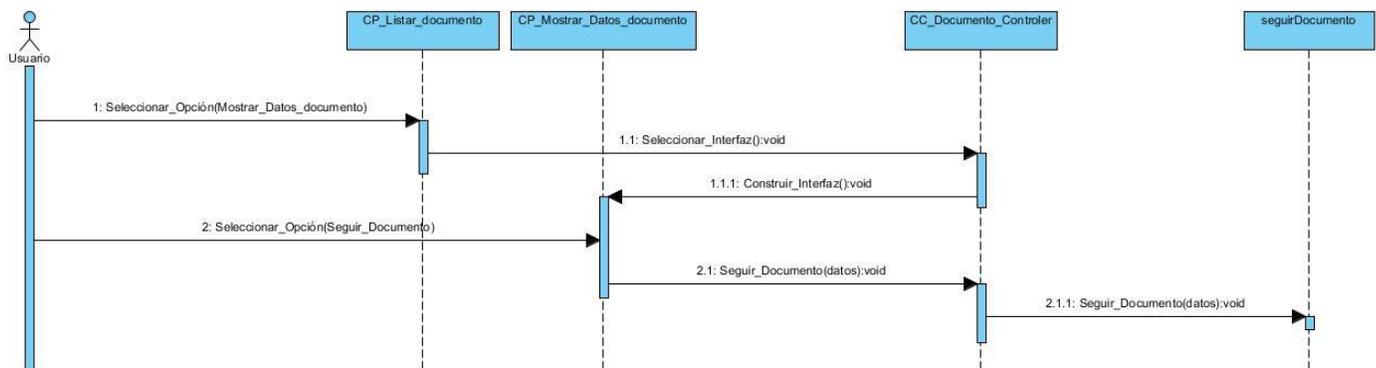


Diagrama de secuencia CU seguir documento.

Anexos

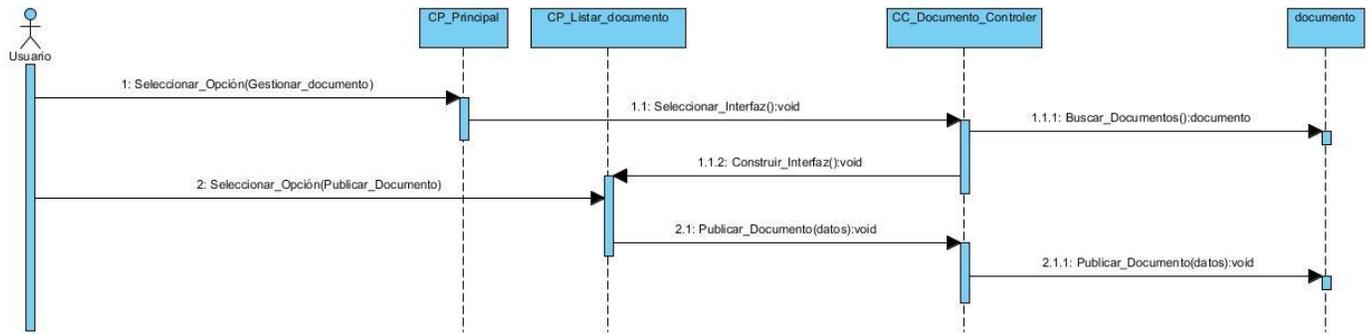


Diagrama de secuencia CU publicar documento.

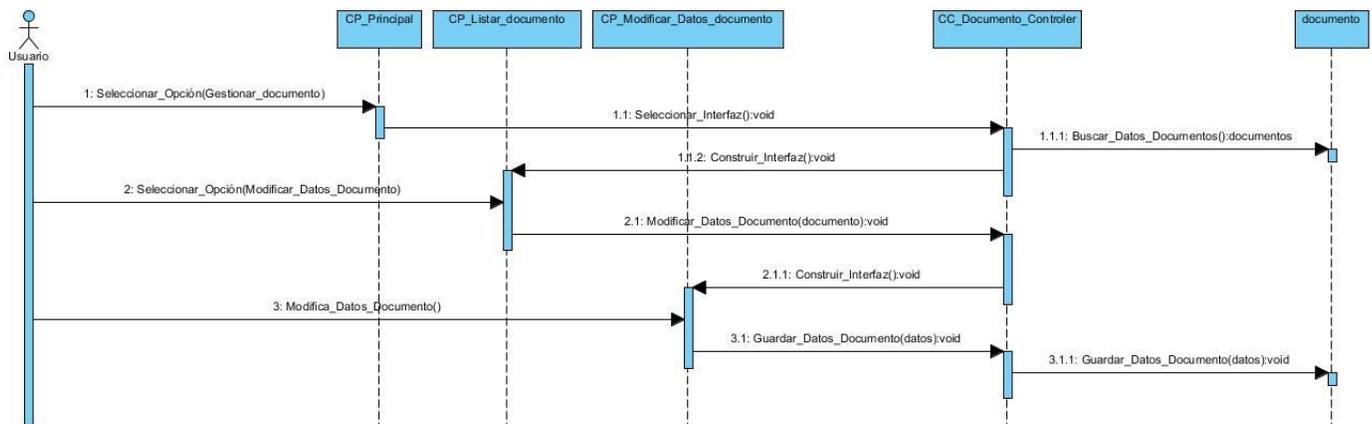


Diagrama de secuencia CU modificar datos documento.

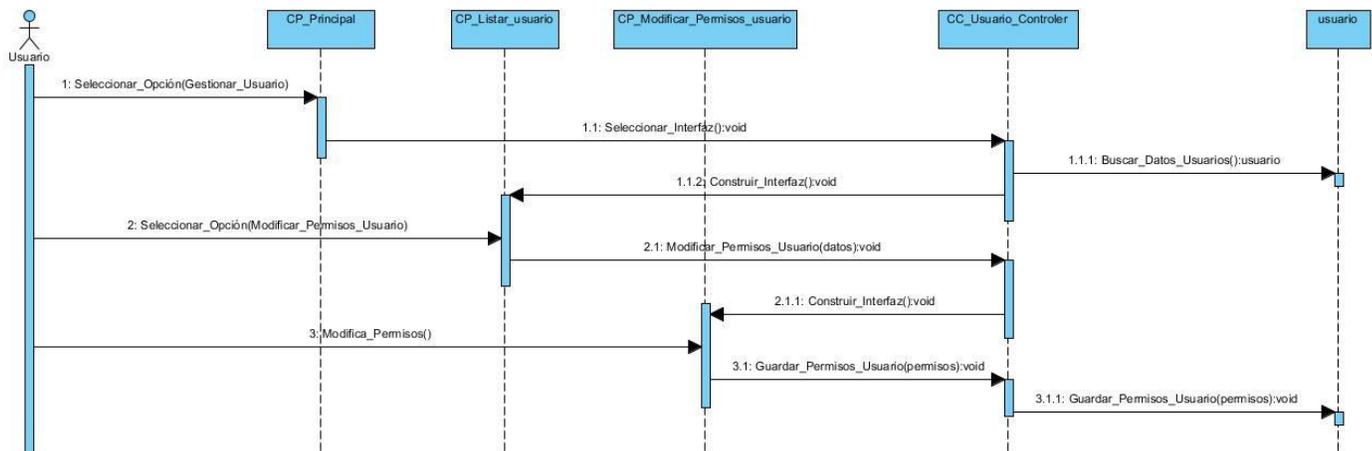


Diagrama de secuencia CU modificar permisos usuario.

Anexos

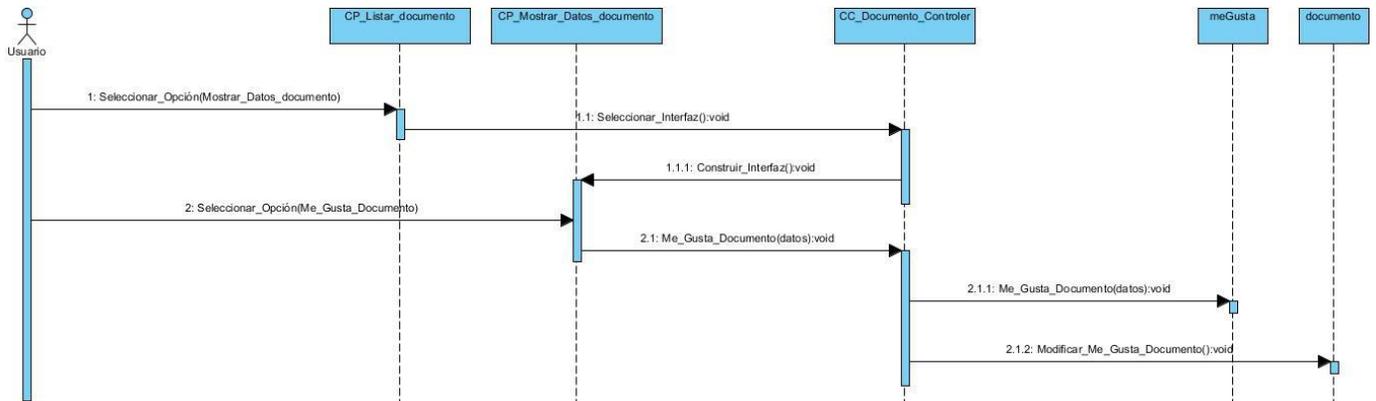


Diagrama de secuencia CU gustar documento.

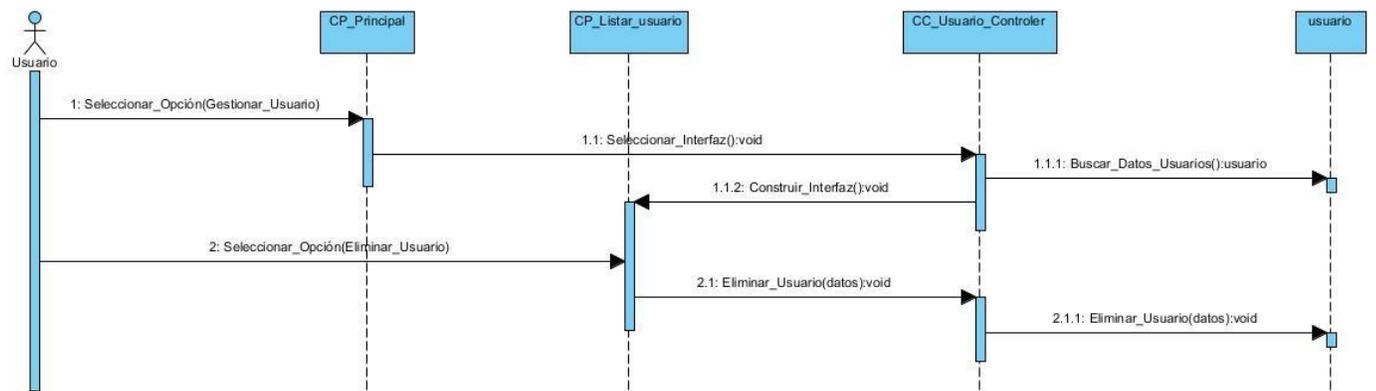


Diagrama de secuencia CU eliminar usuario.

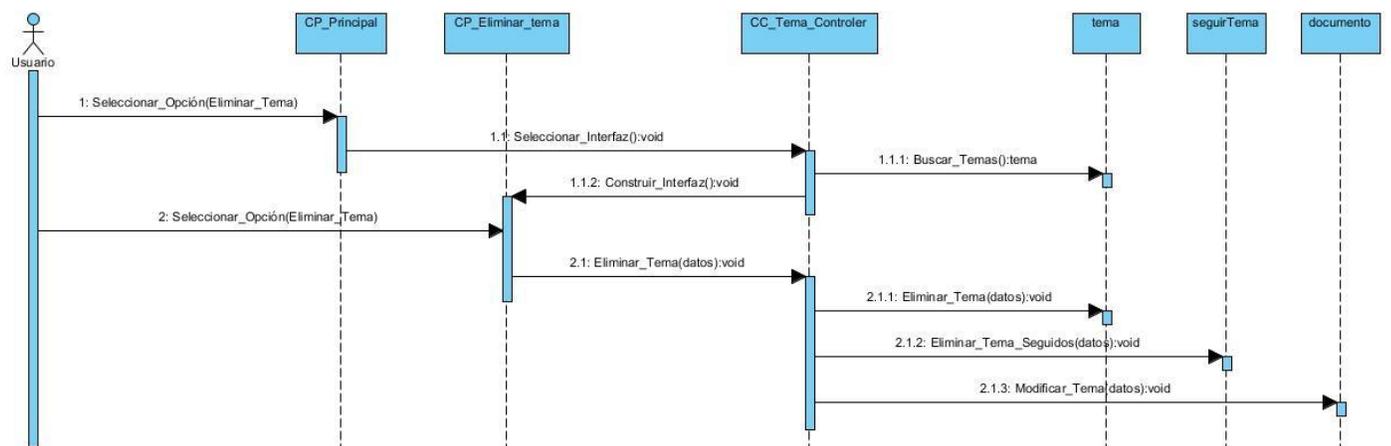


Diagrama de secuencia CU eliminar tema.

Anexos

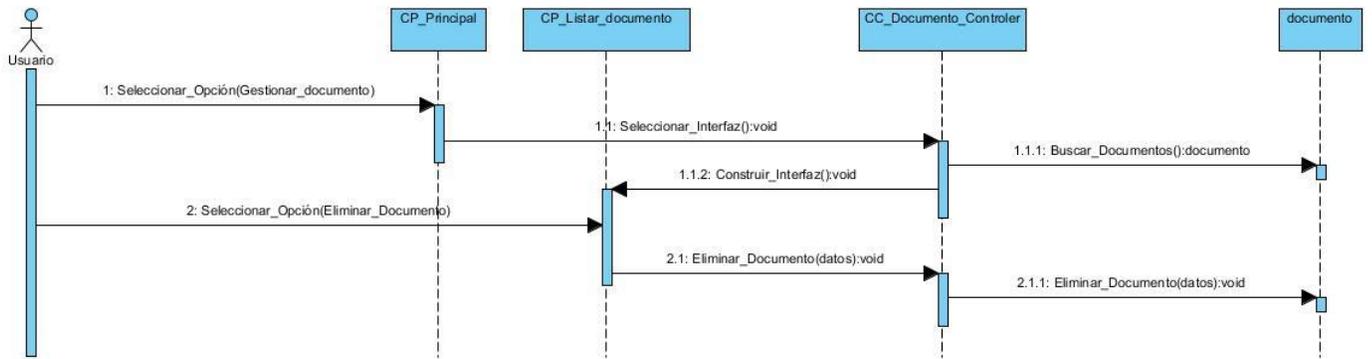


Diagrama de secuencia CU eliminar documento.

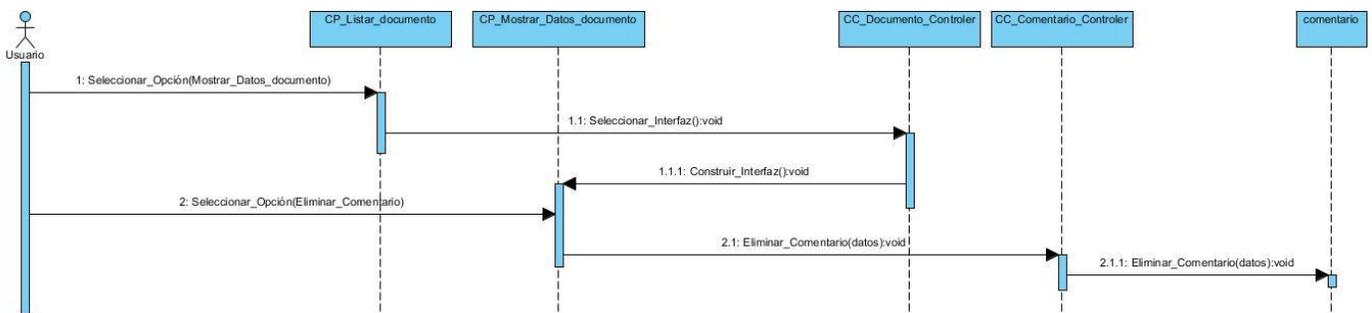


Diagrama de secuencia CU eliminar comentario.

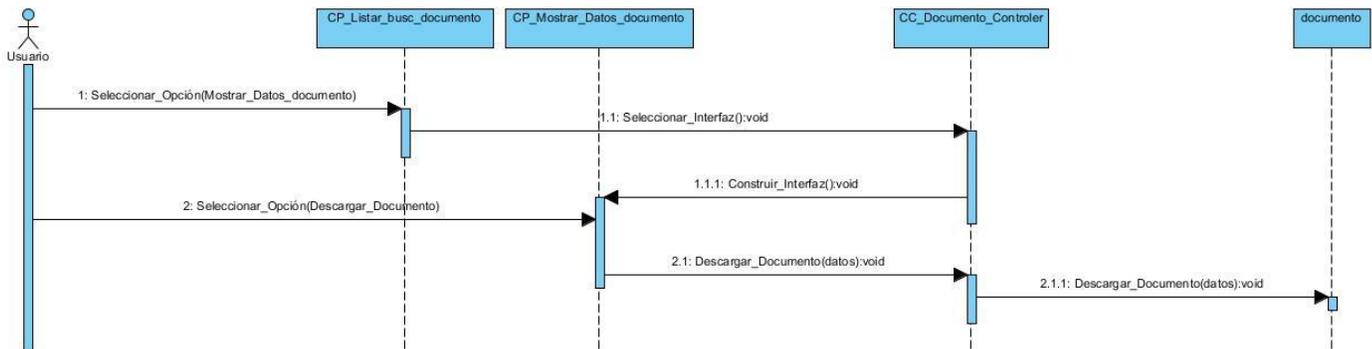


Diagrama de secuencia CU descargar documento.

Anexos

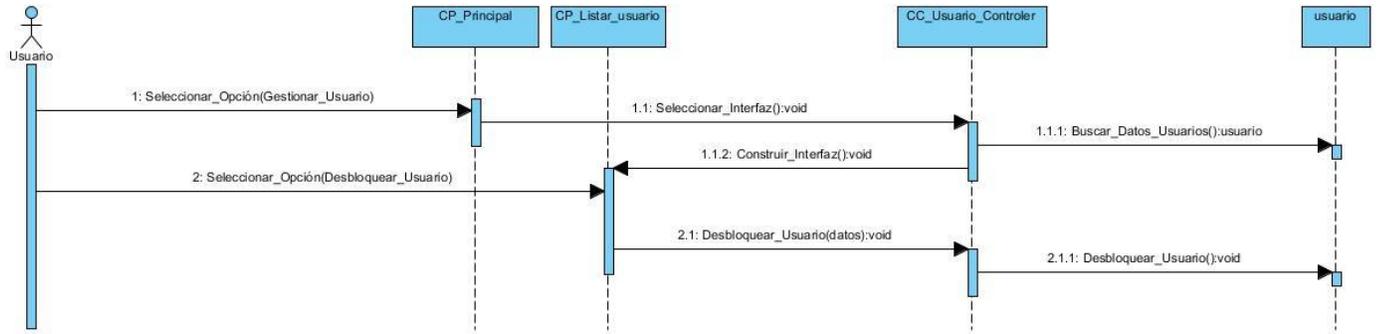


Diagrama de secuencia CU desbloquear usuario.

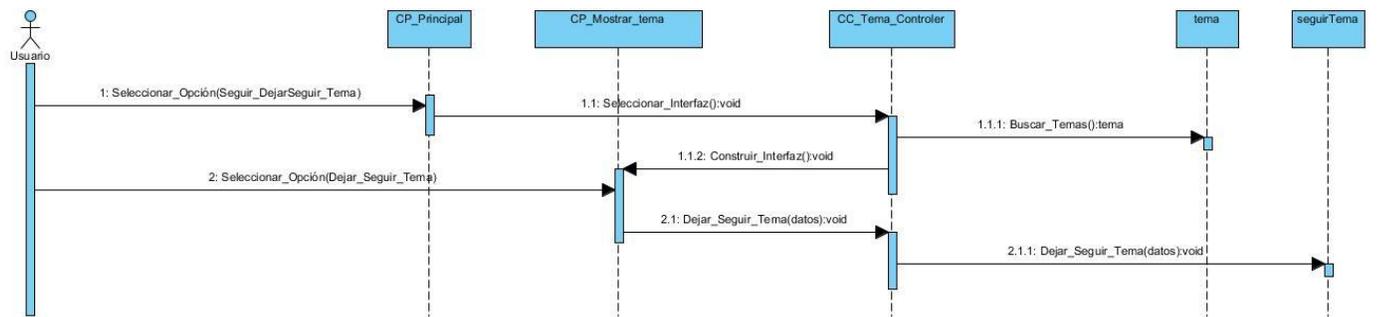


Diagrama de secuencia CU dejar seguir tema.

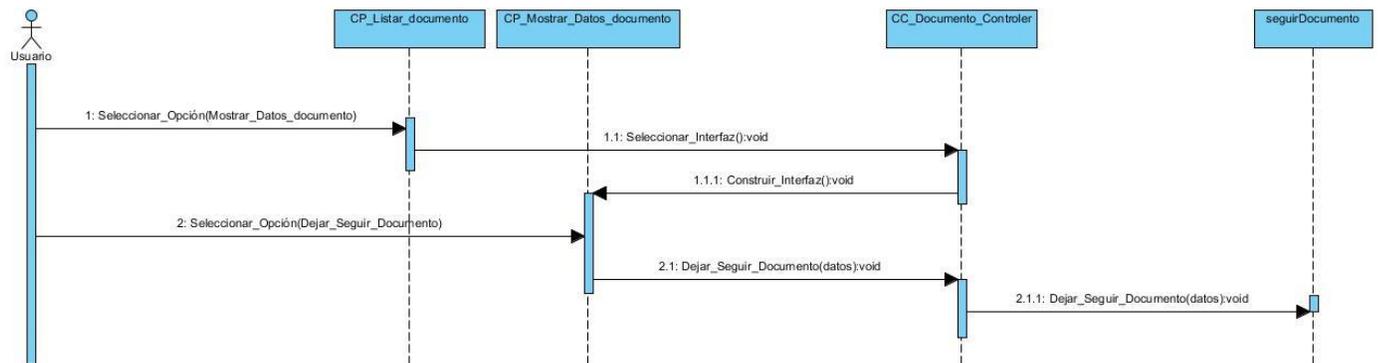


Diagrama de secuencia CU dejar seguir documento.

Anexos

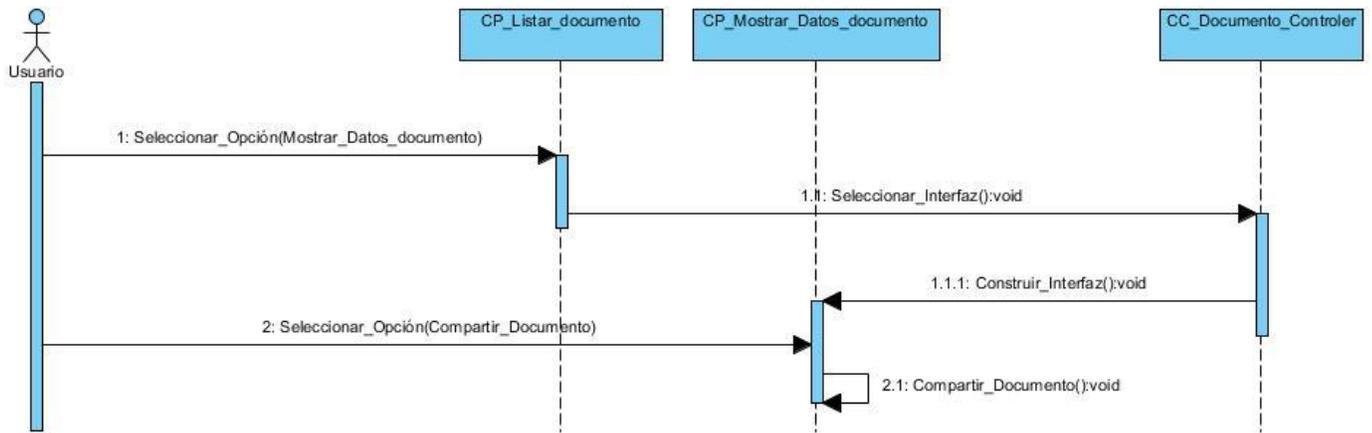


Diagrama de secuencia CU compartir documento.

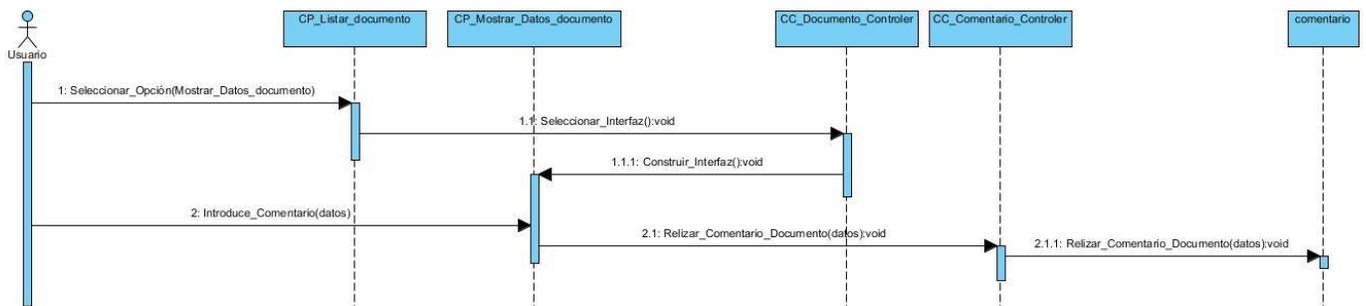


Diagrama de secuencia CU comentar documento.

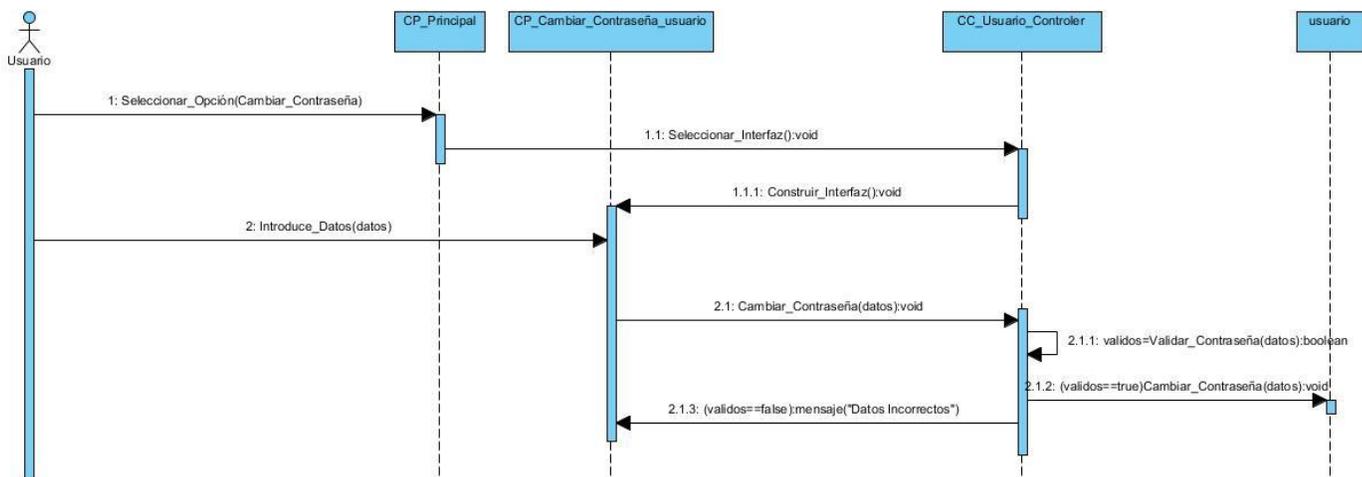


Diagrama de secuencia CU cambiar contraseña.

Anexos

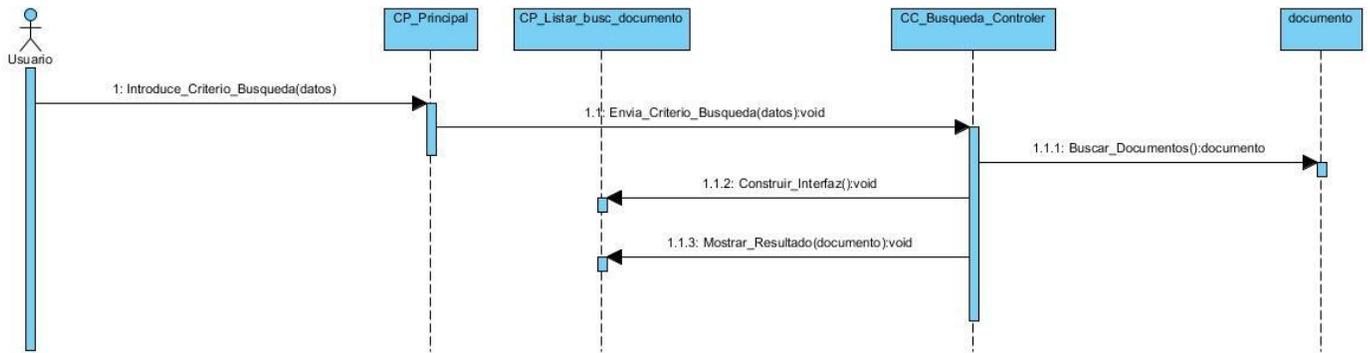


Diagrama de secuencia CU buscar documento.

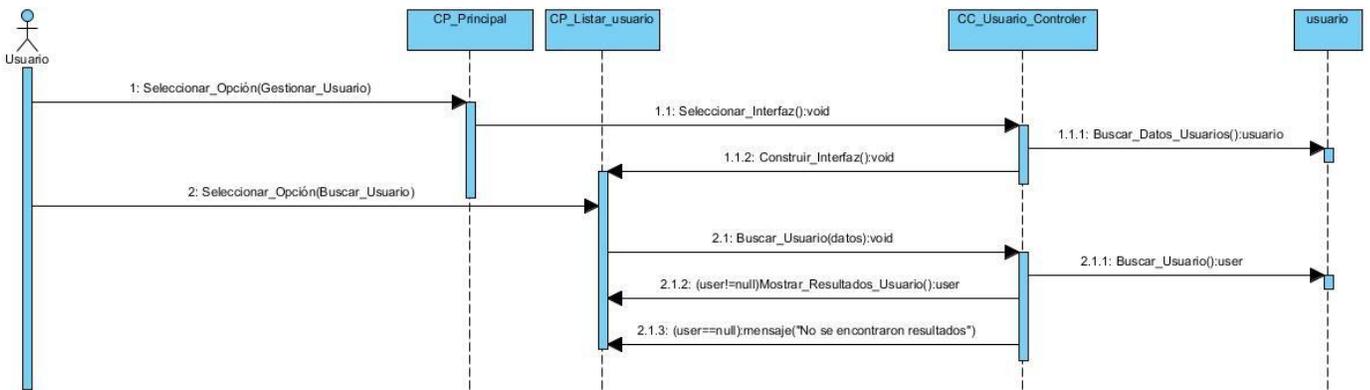


Diagrama de secuencia CU buscar usuario.

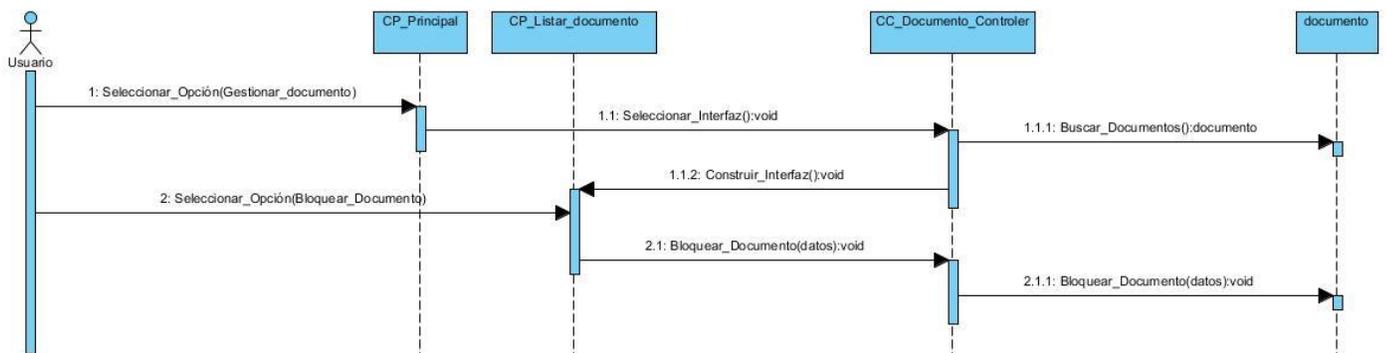


Diagrama de secuencia CU bloquear documento.

Anexos

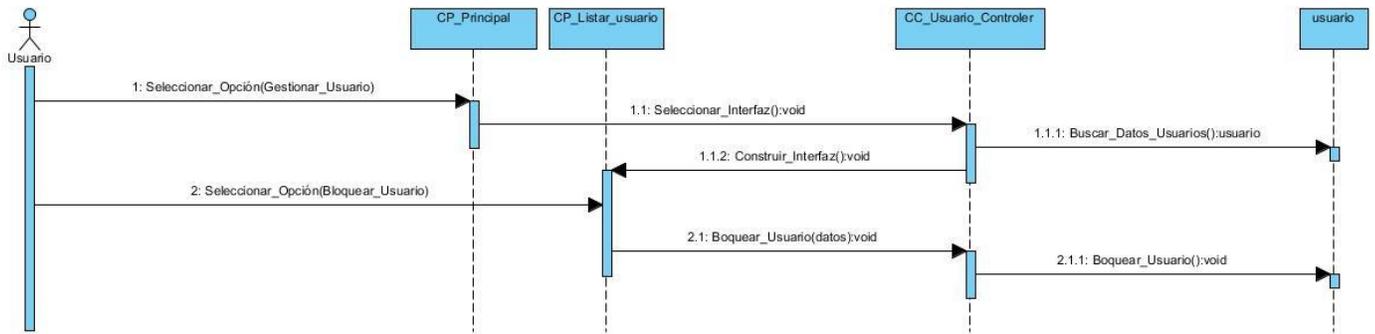


Diagrama de secuencia CU bloquear usuario.

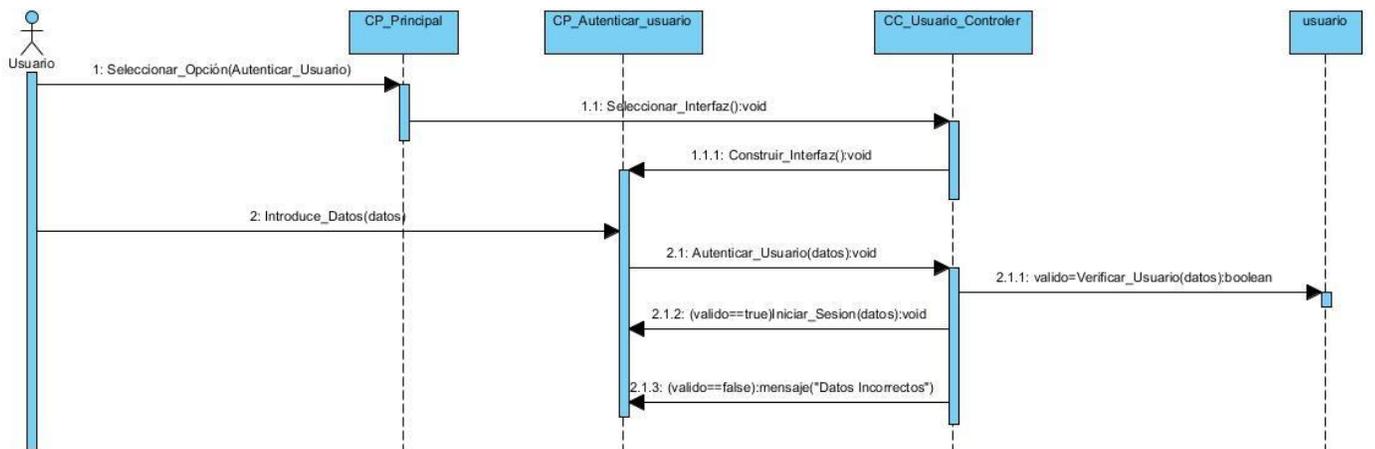


Diagrama de secuencia CU autenticar usuario.

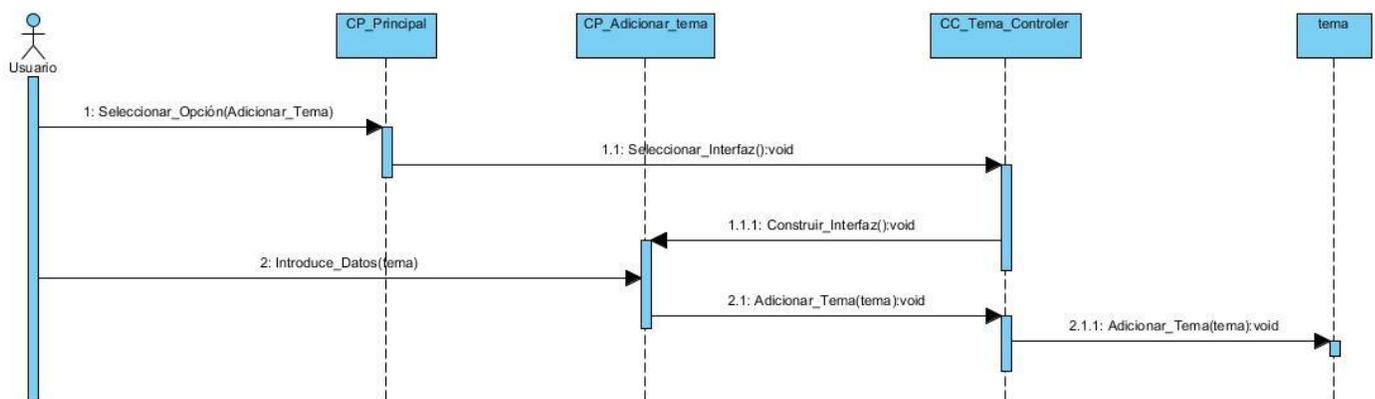


Diagrama de secuencia CU adicionar tema.

Anexos

Anexo 5

[Las celdas de las tablas que contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.]

CU2 Realizar Búsqueda Básica

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	criterio_de_búsqueda	Campo de texto	No	Se debe especificar el criterio por el cual se desea buscar en el título del documento.

Descripción de las variables

Escenario	Descripción	Variable 1: criterio_de_búsqueda	Respuesta del sistema	Flujo central
EC 1.1 Dato correcto y obtiene resultados.	El usuario introduce un criterio correcto en el sistema.	V	El sistema realiza la búsqueda, obtiene resultados y muestra los resultados de la misma.	El usuario introduce el criterio de búsqueda manera correcta y le da clic en Buscar o presiona el botón Enter.
EC 1.2 Campo Vacío	El usuario deja el campo del criterio sin llenar.	I vacía	El sistema muestra un mensaje que dice: " No se encontraron coincidencias".	El usuario deja el campo del criterio sin llenar y le da clic en Buscar o presiona el botón Enter.
EC 1.3 Dato correcto y no obtiene resultados.	El usuario introduce un criterio correcto en el sistema.	V	El sistema realiza la búsqueda, no obtiene resultados y muestra el mensaje: "No se encontraron coincidencias".	El usuario introduce el criterio de búsqueda manera correcta y le da clic en Buscar o presiona el botón Enter.

Caso de prueba del CU Realizar Búsqueda Básica

CU2 Descargar Documento

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Descarga el documento.	El usuario selecciona del documento el ícono de descarga.	El sistema le permite al usuario descargar el documento en su computadora.	El usuario da clic en el ícono de descarga.

Caso de prueba del CU Descargar Documento

Anexos

CU3 Autenticar Usuario

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	usuario	Campo de texto	No	Se debe especificar un usuario existente en el sistema.
2	contraseña	Campo de texto	No	Se debe especificar la contraseña perteneciente al usuario especificado.

Descripción de las variables

Escenario	Descripción	Variable 1: usuario	Variable 2: contraseña	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos.	El usuario introduce el usuario y la contraseña de manera correcta en el sistema.	V	V	El sistema inicia la sesión y redirecciona a la página de inicio.	El usuario introduce los dos campos de manera correcta y le da clic en Aceptar o presiona el botón Enter.
EC 1.2 Campos vacíos.	El usuario deja los dos campos sin llenar.	I	I	El sistema muestra un mensaje que dice: "Datos incorrectos.".	El usuario deja algún campo de vacío y le da clic en Aceptar o presiona el botón Enter.
EC 1.3 Dato incorrecto.	El usuario introduce un valor incorrecto en el sistema.	I	V	El sistema muestra un mensaje que dice: "Datos incorrectos.".	El usuario introduce un campo de manera incorrecta y le da clic en Aceptar o presiona el botón Enter.
		V	I	El sistema muestra un mensaje que dice: "Datos incorrectos.".	El usuario introduce un campo de manera incorrecta y le da clic en Aceptar o presiona el botón Enter.

Caso de prueba del CU Autenticar Usuario

CU4 Realizar Búsqueda Avanzada

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	criterio_de_búsqueda	Campo de texto	No	Se debe especificar el criterio por el cual se desea buscar en el título del documento.
2	autor	Campo de texto	Si	Se debe especificar al menos una palabra y sin caracteres extraños (no es obligatorio para realizar la búsqueda).

Anexos

3	tema	lista de opciones	Si	Se debe seleccionar una opción (no es obligatorio para realizar la búsqueda).
4	tipo	lista de opciones	Si	Se debe seleccionar una opción (no es obligatorio para realizar la búsqueda).
5	lenguaje	lista de opciones	Si	Se debe seleccionar una opción (no es obligatorio para realizar la búsqueda).

Descripción de las variables

Escenario	Descripción	Variable 1: criterio_de_búsqueda	Variable 2: autor	Variable 3: tema	Variable 4: tipo	Variable 5: lenguaje	Respuesta del sistema	Flujo central
EC 1.1 Dato correcto y obtiene resultados.	El usuario introduce un criterio correcto en el sistema.	V	N/A	N/A	N/A	N/A	El sistema realiza la búsqueda, obtiene resultados y muestra los resultados de la misma.	El usuario introduce el criterio de búsqueda manera correcta y le da clic en Buscar o presiona el botón Enter.
EC 1.2 Campo Vacío	El usuario deja el campo del criterio sin llenar.	I vacía	N/A	N/A	N/A	N/A	El sistema muestra un mensaje que dice: " No se encontraron coincidencias" .	El usuario deja el campo del criterio sin llenar y le da clic en Buscar o presiona el botón Enter.
EC 1.3 Dato correcto y no obtiene resultados.	El usuario introduce un criterio correcto en el sistema.	V	N/A	N/A	N/A	N/A	El sistema realiza la búsqueda, no obtiene resultados y muestra el mensaje: "No se encontraron coincidencias" .	El usuario introduce el criterio de búsqueda manera correcta y le da clic en Buscar o presiona el botón Enter.

Caso de prueba del CU Realizar Búsqueda Avanzada

Anexos

CU6 Comentar Documento

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	comentario	campo de texto	No	Se debe introducir una cadena de caracteres.

Descripción de la variable

Escenario	Descripción	Variable 1: comentario	Respuesta del sistema	Flujo central
EC 1.1 Dato correcto	El usuario introduce un comentario de manera correcta en el sistema.	V	El sistema adiciona el comentario al sistema.	El usuario introduce un comentario de manera correcta y le da clic en Comentar.
EC 1.2 Campo Vacío	El usuario deja el campo sin llenar.	I vacía	El sistema muestra un mensaje: "Debe de escribir un comentario.".	El usuario deja el campo vacío y le da clic en Comentar.

Caso de prueba del CU Comentar Documento

CU7 Compartir Documento

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Comparte el documento.	El usuario selecciona del documento el ícono de la red social en donde desea compartir el documento.	El sistema le permite al usuario compartir el documento en la red social.	El usuario selecciona del documento el ícono de la red social en donde desea compartir el documento dando clic en la misma.

Caso de prueba del CU Compartir Documento

CU8 Votar Documento

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	voto	grupo de opciones	No	Se debe seleccionar un valor para proporcionar.

Descripción de la variable

Escenario	Descripción	Variable 1: voto	Respuesta del sistema	Flujo central
EC 1.2 Dato correcto	El usuario selecciona el valor que proporcionará al documento en el sistema (valor de 1-5).	N/A	El sistema registra el voto realizado y no permite que este usuario vote de nuevo por ese documento.	El usuario selecciona el valor que proporcionará al documento (de no seleccionar ningún valor por defecto asume 1) en el sistema y le da clic en Votar.

Caso de prueba del CU Votar Documento

Anexos

CU9 Gustar Documento

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opinión, si gusta el documento.	El usuario selecciona del documento la opción "Me Gusta".	El sistema registra el criterio del usuario sobre el documento en el sistema.	El usuario selecciona del documento la opción "Me Gusta" dando clic en la misma.

Caso de prueba del CU Gustar Documento

CU10 Seguir Documento

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Seguir el documento.	El usuario selecciona del documento la opción "Seguir".	El sistema hace seguidor al usuario del documento, de manera que envía reportes de actividad sobre este documento al correo del usuario.	El usuario selecciona del documento la opción "Seguir" dando clic en la misma.

Caso de prueba del CU Seguir Documento

CU11 Seguir Tema

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	tema	lista de opciones	No	Se debe seleccionar un tema.

Descripción de la variable

Escenario	Descripción	Variable 1: tema	Respuesta del sistema	Flujo central
EC 1.1 Sin selección.	El usuario no selecciona ninguna opción.	I	El sistema muestra un mensaje: "Debe seleccionar un tema.".	El usuario no selecciona ningún tema y le da clic en Seguir o presiona el botón Enter.
		vacía		
EC 1.2 Dato correcto	El usuario selecciona un tema de los que proporciona el sistema.	V	El sistema hace seguidor al usuario del tema seleccionado, de manera que envía reportes de actividad sobre este tema al correo del usuario.	El usuario selecciona el tema que desea ser seguidor y le da clic en Seguir o presiona el botón Enter.

Caso de prueba del CU Seguir Tema

CU12 Modificar Permisos Usuarios

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	comentar_documento	Opción de selección	Si	Se selecciona si se desea otorgar este permiso, en caso contrario no se selecciona.

Anexos

2	subir_documento	Opción de selección	Si	Se selecciona si se desea otorgar este permiso, en caso contrario no se selecciona.
3	rol_admin	Opción de selección	Si	Se selecciona si se desea otorgar este permiso, en caso contrario no se selecciona.
4	rol_superadmin	Opción de selección	Si	Se selecciona si se desea otorgar este permiso, en caso contrario no se selecciona.

Descripción de las variables

Escenario	Descripción	Variable 1: comentar_documento	Variable 2: subir_documento	Variable 3: rol_admin	Variable 4: rol_superadmin	Respuesta del sistema	Flujo central
EC 1.1 Modifica permisos usuario.	El administrador selecciona los permisos que desea que tenga el usuario en el sistema.	N/A	N/A	N/A	N/A	El sistema modifica los permisos del usuario y regresa a la página que muestra la lista con todos los usuarios.	El administrador selecciona los permisos que desea que tenga el usuario en el sistema y le da clic en Guardar o presiona el botón Enter.

Caso de prueba del CU Modificar Permisos Usuario

CU13 Bloquear Usuario

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Bloquea el usuario.	El administrador selecciona de las acciones de un usuario la acción "Bloquear".	El sistema bloquea al usuario de manera que no puede autenticarse en el sistema.	El administrador selecciona de las acciones de un usuario la acción "Bloquear" dando clic en la misma.

Caso de prueba del CU Bloquear Usuario

CU14 Publicar Documento

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Publica el documento.	El administrador selecciona de las acciones de un documento la acción "Publicar".	El sistema publica el documento de manera que ya aparece en las búsquedas del sistema para los usuarios.	El administrador selecciona de las acciones de un documento la acción "Publicar" dando clic en la misma.

Caso de prueba del CU Publicar Documento

Anexos

CU15 Eliminar Comentario

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Elimina un comentario.	El administrador selecciona de un comentario sobre un documento el ícono de eliminar.	El sistema elimina el comentario del sistema.	El administrador selecciona de un comentario sobre un documento ícono de eliminar dando clic en el mismo.

Caso de prueba del CU Eliminar Comentario

CU16 Eliminar Usuario

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Elimina un usuario.	El administrador selecciona de las acciones de un usuario la acción "Eliminar".	El sistema elimina al usuario del sistema.	El administrador selecciona de las acciones de un usuario la acción "Eliminar" dando clic en la misma.

Caso de prueba del CU Eliminar Usuario

CU17 Mostrar Información Documento

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Muestra datos de documento.	El usuario selecciona de la lista de documentos encontrados en una búsqueda el documento que desee ver los datos asociados al mismo dando clic sobre el mismo.	El sistema muestra una nueva interfaz con los datos asociados al documento y algunas opciones.	El usuario selecciona de la lista de documentos encontrados en una búsqueda el documento que desee ver los datos asociados al mismo dando clic sobre el mismo.

Caso de prueba del CU Mostrar Información Documento

CU20 Confirmar Usuario

EC 1.1 Confirma un nuevo usuario.	El usuario selecciona del correo que le fue enviado al registrarse en el sistema el enlace que este posee dando clic en el y el sistema confirma el usuario en el sistema.	El sistema confirma al usuario en el sistema mostrando y muestra un mensaje de confirmación en la página donde fue redireccionado por el link del correo.	El usuario selecciona del correo que le fue enviado al registrarse en el sistema el enlace que este posee dando clic en él.
-----------------------------------	--	---	---

Caso de prueba del CU Confirmar Usuario

CU21 Bloquear Documento

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Bloquea el documento.	El administrador selecciona de las acciones de un documento la acción "Bloquear".	El sistema bloquea el documento de manera que no aparece en las búsquedas del sistema para los usuarios.	El administrador selecciona de las acciones de un documento la acción "Bloquear" dando clic en la misma.

Caso de prueba del CU Bloquear Documento

Anexos

CU22 Modificar Datos Documento

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	título	Campo de texto	No	Se debe especificar al menos una palabra y sin caracteres extraños.
2	autor	Campo de texto	No	Se debe especificar al menos una palabra y sin caracteres extraños.
3	tema	lista de opciones	No	Se debe seleccionar un tema.
4	tipo	lista de opciones	No	Se debe seleccionar un tipo.
5	lenguaje	lista de opciones	No	Se debe seleccionar un lenguaje.

Descripción de las variables

Escenario	Descripción	Variable 1: título	Variable 2: autor	Variable 3: tema	Variable 4: tipo	Variable 5: lenguaje	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos.	El usuario introduce todos los datos de manera correcta en el sistema.	V	V	V	V	V	El sistema modifica los datos del documento en el sistema y muestra la página principal.	El usuario llena todos los campos de manera correcta y le da clic en Modificar o presiona el botón Enter.
EC 1.2 Campos Vacíos	El usuario deja algunos de los campos, título o autor sin llenar.	I	I	N/A	N/A	N/A	Si el navegador es Firefox a partir de la versión 5 el sistema muestra un mensaje que dice: "Por favor, rellene este campo" o "Please fill out this field" en dependencia del idioma que tenga el navegador.	El usuario deja algunos de los campos título o autor sin llenar y le da clic en Modificar o presiona el botón Enter.
		vacía	vacía					

Caso de prueba del CU Modificar Datos Documento

CU23 Eliminar Documento

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Elimina un documento.	El administrador selecciona de las acciones de un documento la acción "Eliminar".	El sistema elimina el documento del sistema.	El administrador selecciona de las acciones de un documento la acción "Eliminar" dando clic en la misma.

Caso de prueba del CU Eliminar Documento

Anexos

CU25 Buscar Usuario

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	usuario	Campo de texto	No	Se debe especificar el usuario que se desea buscar en el sistema.

Descripción de la variable

Escenario	Descripción	Variable 1: usuario	Respuesta del sistema	Flujo central
EC 1.1 Dato correcto y obtiene resultados.	El administrador introduce un usuario que existe en el sistema.	V	El sistema realiza la búsqueda, obtiene resultados y muestra los resultados de la misma.	El administrador introduce el usuario que desea buscar de manera correcta y le da clic en Buscar Usuario o presiona el botón Enter.
EC 1.2 Campo Vacío	El administrador deja el campo del criterio sin llenar.	I vacía	El sistema muestra todos los usuarios del sistema.	El administrador deja el campo del criterio sin llenar y le da clic en Buscar Usuario o presiona el botón Enter.
EC 1.3 Dato incorrecto.	El administrador introduce un usuario que no existe en el sistema.	I	El sistema muestra el mensaje: "No se encontraron coincidencias".	El administrador introduce un usuario que no existe en el sistema y le da clic en Buscar Usuario o presiona el botón Enter.

Caso de prueba del CU Buscar Usuario

CU26 Cambiar Contraseña

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	contraseña_actual	Campo de texto	No	Se debe especificar la contraseña perteneciente al usuario que está iniciada su sesión.
2	contraseña_nueva	Campo de texto	No	Se debe especificar una secuencia de caracteres que sea igual a la de la variable 3.
3	repetir_contraseña	Campo de texto	No	Se debe especificar una secuencia de caracteres que sea igual a la de la variable 2.

Descripción de las variables

Escenario	Descripción	Variable 1: contraseña_actual	Variable 2: contraseña_nueva	Variable 3: repetir_contraseña	Respuesta del sistema	Flujo central
EC 1.1	El usuario	V	V	V	El sistema cambia la	El usuario llena

Anexos

Datos correctos	introduce todos los valores de manera correcta en el sistema.				contraseña del usuario en el sistema y muestra un mensaje informando: "Se ha cambiado la contraseña correctamente" y muestra el usuario y el correo del mismo.	todos los campos de manera correcta y le da clic en Cambiar Contraseña o presiona el botón Enter.
EC 1.2 Campos Vacíos	El usuario deja todos o algunos campos sin llenar.	I	I	I	Si el navegador es Firefox a partir de la versión 5 el sistema muestra un mensaje que dice: "Por favor, rellene este campo" o "Please fill out this field" en dependencia del idioma que tenga el navegador.	El usuario deja algún campo de vacío y le da clic en Cambiar Contraseña o presiona el botón Enter.
		vacía	vacía	vacía		
EC 1.3 Dato incorrecto.	El usuario introduce un valor incorrecto en el sistema.	I	V	V	El sistema muestra un mensaje: "La contraseña no es correcta.".	El usuario llena todos los campos de manera correcta menos su contraseña y le da clic en Cambiar Contraseña o presiona el botón Enter.
		V	I	V		
		V	V	I	El sistema muestra un mensaje: "La contraseña no coincide".	El usuario llena todos los campos de manera correcta pero no coinciden las dos contraseñas que ingresó (variable 2 y 3) y le da clic en Cambiar Contraseña o presiona el botón Enter.
		V	V	I		

Caso de prueba del CU Cambiar Contraseña

Anexos

CU27 Dejar Seguir Documento

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Dejar de seguir el documento.	El usuario selecciona del documento la opción "Seguir".	El sistema quita al usuario de seguidor del documento, de manera que elimina el envío de reportes al correo del usuario.	El usuario selecciona del documento la opción "Dejar de Seguir" dando clic en la misma.

Caso de prueba del CU Dejar Seguir Documento

CU28 Dejar Seguir Tema

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	tema	lista de opciones	No	Se debe seleccionar un tema.

Descripción de la variable

Escenario	Descripción	Variable 1: tema	Respuesta del sistema	Flujo central
EC 1.1 Sin selección.	El usuario no selecciona ninguna opción.	I	El sistema muestra un mensaje: "Debe seleccionar un tema.".	El usuario no selecciona ningún tema y le da clic en Seguir o presiona el botón Enter.
		vacía		
EC 1.2 Dato correcto	El usuario selecciona un tema de los que proporciona el sistema.	V	El sistema quita al usuario de seguidor del tema seleccionado, de manera que elimina el envío de reportes al correo del usuario.	El usuario selecciona el tema que desea dejar de ser seguidor y le da clic en Dejar de Seguir o presiona el botón Enter.

Caso de prueba del CU Dejar Seguir Tema

CU29 Adicionar Tema

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	tema	Campo de texto	No	Se debe especificar al menos una palabra y sin caracteres extraños ni números.

Descripción de la variable

Escenario	Descripción	Variable 1: tema	Respuesta del sistema	Flujo central
EC 1.1 Dato correcto	El usuario introduce el valor de manera correcta en el sistema.	V	El sistema adiciona el tema al sistema y redirecciona a la página principal.	El usuario introduce el dato de manera correcta y le da clic en Adicionar o presiona el botón Enter.
EC 1.2 Campo	El usuario deja el	I	Si el navegador es Firefox a	El usuario deja el campo vacío

Anexos

Vacío	campo sin llenar.	vacía	partir de la versión 5 el sistema muestra un mensaje que dice: "Por favor, rellene este campo" o "Please fill out this field" en dependencia del idioma que tenga el navegador.	y le da clic en Adicionar o presiona el botón Enter.
EC 1.3 Dato incorrecto.	El usuario introduce un tema que ya existe en el sistema.	I	El sistema muestra un mensaje: "El tema ya existe."	El usuario introduce el tema de manera correcta pero el mismo ya existe en el sistema y le da clic en Adicionar o presiona el botón Enter.

Caso de prueba del CU Adicionar Tema

CU30 Eliminar Tema

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	tema	lista de opciones	No	Se debe seleccionar un tema.

Descripción de la variable

Escenario	Descripción	Variable 1: tema	Respuesta del sistema	Flujo central
EC 1.1 Sin selección.	El usuario no selecciona ninguna opción.	I vacía	El sistema muestra un mensaje: "Debe seleccionar un tema."	El usuario no selecciona ningún tema y le da clic en Eliminar o presiona el botón Enter.
EC 1.2 Dato correcto	El usuario selecciona un tema de los que proporciona el sistema.	V	El sistema elimina el tema del sistema, en todos los documentos donde el tema sea igual al eliminado cambia el valor por "otros" y redirecciona a la página principal.	El usuario selecciona el tema que desea eliminar y le da clic en Eliminar o presiona el botón Enter.

Caso de prueba del CU Eliminar Tema

CU31 Desbloquear Usuario

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Desbloquea el usuario.	El administrador selecciona de las acciones de un usuario la acción "Desbloquear".	El sistema desbloquea al usuario de manera que ya puede autenticarse en el sistema.	El administrador selecciona de las acciones de un usuario la acción "Desbloquear" dando clic en la misma.

Caso de prueba del CU Desbloquear Usuario

Anexos

Anexo 6

	Total
aggregate_report_count(total de peticiones)	2060
Average(Tiempo promedio de respuesta en ms)	4572
aggregate_report_median(Mediana del tiempo de respuesta)	5342
aggregate_report_min(Tiempo de respuesta mínimo)	2
aggregate_report_max(Tiempo de respuesta máximo)	19973
aggregate_report_error%(Errores detectados en %)	0.0014563106796116505

Informe resumen elaborado por la herramienta Jmeter.