

Aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0

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

Autores:

Maylen Xiomara Benítez Ros
Lázaro Ricardo Rodríguez Placeres

Tutores:

Ing. Yonny Mondelo Hernández
Ing. Yasel Almenares Alfonso

La Habana, 2013

A faded, grayscale portrait of Fidel Castro, showing his face and upper torso. He is wearing a dark suit jacket over a light-colored shirt and a dark tie. The image is semi-transparent, allowing the text to be overlaid on it.

“Revolución es sentido del momento histórico; es cambiar todo lo que debe ser cambiado; es igualdad y libertad plenas; es ser tratado y tratar a los demás como seres humanos; es emanciparnos por nosotros mismos y con nuestros propios esfuerzos; es desafiar poderosas fuerzas dominantes dentro y fuera del ámbito social y nacional; es defender valores en los que se cree al precio de cualquier sacrificio; es modestia, desinterés, altruismo, solidaridad y heroísmo; es luchar con audacia, inteligencia y realismo; es no mentir jamás ni violar principios éticos; es convicción profunda de que no existe fuerza en el mundo capaz de aplastar la fuerza de la verdad y las ideas. Revolución es unidad, es independencia, es luchar por nuestros sueños de justicia para Cuba y para el mundo, que es la base de nuestro patriotismo, nuestro socialismo y nuestro internacionalismo.”

Fidel Castro Ruz.

DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores de la presente tesis y autorizamos al Centro CIDI de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2013.

Autor: Maylen Xiomara Benítez Ros

Autor: Lázaro Ricardo Rodríguez
Placeres

Tutor: Ing. Yonny Mondelo
Hernández

Tutor: Ing. Yasel Almenares Alfonso

DEDICATORIA

*A mi vieja.
A mi familia: Mimi, Pato, Churry, Mami.
Maylen.*

*A mi mamá, mi abuela y mi padrastro que me dieron lo que
tenían para que fuese universitario.
Lázaro*

AGRADECIMIENTOS

Gracias a todos los que tuvieron que ver con lo que soy hoy.

Gracias a mis viejas, que soportaron tantas malacrianzas.

Gracias al hombre más importante: mi papá.

Gracias a la maestra más dura: mi mamá.

Gracias a las mejores cómplices: Estelita y Sari.

Gracias a los que molesté tantas veces: tío Sergio, tío Nene y tía Aída.

Gracias a mis amigas: Delmis, Maricet, Yanelis, Taimí y María, no lo habría logrado sin su compañía.

Gracias a mi dolor más grande: "papi".

Gracias a mis tutores: Gilberto y Yasel.

Gracias a Dios por haberlos puesto a todos en mi vida.

Maylen.

A Madé y Mayí, las dos que siempre estaban aunque no hicieran falta.

A YannierP y Luis Daniel (mi maestro) que me sacaron de los aprietos más grandes cuando no ponía una.

A Yasiel y Yadiel, que siempre me ayudaron.

A Yanelis, Marialeisy, Yordany, Liubert y Barbán.

A Yasel, mi tutor.

A mis profesores del proyecto Elieyis, Yuneisy y Gilberto.

Gracias MAYLEN por ser mi compañera de tesis.

Lázaro.

RESUMEN

A pesar de la importancia que han adquirido los nuevos formatos de almacenamiento de información, actualmente persisten las dificultades en la gestión adecuada de los archivos de audio y vídeo digitales. El tratamiento de estos recursos se dificulta debido al empleo de rudimentarios sistemas de organización que impiden su uso óptimo. Como consecuencia de que el Sistema de Gestión de Medias para la Prensa desarrollado en la UCI, para la gestión de audio y vídeos digitales, presenta numerosas desventajas que impiden un desempeño óptimo del proceso de gestión documental, se desarrolló una aplicación web para dicho sistema que mejora la usabilidad y facilita el trabajo del usuario. El proceso de desarrollo de software fue completamente guiado por la metodología OpenUP, apoyado en tecnologías como los *framework* Symfony 2.1.4, jQueryUI 1.9.2 y Bootstrap 2.2.2; el sistema gestor de base de datos PostgreSQL 9.1 y Visual Paradigm 8.0 como herramienta para el modelado. La aplicación desarrollada: TeVeo Plus V1.0, reúne un conjunto de características que la convierten en una solución apta para resolver problemas de tratamiento, mantenimiento y conservación de archivos de audio y vídeos en las unidades de información de entidades cuyas actividades diarias dependen de la gestión de estos recursos.

Palabras clave: audio, ECM Alfresco, gestión documental, usabilidad, vídeos.

ÍNDICE

INTRODUCCIÓN1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 6

 1.1. Revisión conceptual de la propuesta de solución6

 1.1.1 Gestión documental.....6

 1.1.2 Mediatecas..... 7

 1.1.3 Audio digital7

 1.1.4. Vídeo digital8

 1.1.5. Gestores de Contenidos Empresariales (ECM).....8

 1.1.6. Características del ECM Alfresco9

 1.1.7. Usabilidad9

 1.1.8. Accesibilidad.....10

 1.2. Tendencias actuales de las interfaces web desarrolladas para el manejo de archivos multimedia...10

 1.2.1. YouTube.....11

 1.2.2. Vimeo.....11

 1.2.3. teVeo.....12

 1.2.4. Primicia.....12

 1.2.5. Observaciones13

 1.3. Tendencias actuales de las interfaces web desarrolladas para el ECM Alfresco14

 1.3.1. Alfresco Explorer.....14

 1.3.2. Opsoro.....15

 1.3.3. iFresco.....15

 1.3.4. eXcriba16

 1.3.5. Observaciones17

 1.4. Marcos de trabajo (*framework*).....17

 1.4.1. Symfony 2.1.418

1.4.2. jQueryUI 1.9.2	18
1.4.3. Bootstrap 2.2.2.....	18
1.4.4. Justificación de la selección.....	18
1.5. Lenguajes para la Web	19
1.5.1. PHP 5.3.10.....	19
1.5.2. HTML 5.....	20
1.5.3. CSS 3	20
1.5.4. JavaScript	20
1.5.5. Justificación de la selección.....	20
1.6. Servidor <i>streaming</i>	21
1.6.1. Flumotion 0.10.0	21
1.7. Entornos Integrados de Desarrollo (IDE)	22
1.7.1. NetBeans 7.3.....	22
1.8. Sistemas Gestores de Bases de Datos (SGBD).....	22
1.8.1. PostgreSQL 9.1.....	22
1.9. Ingeniería del Software asistida por computadoras (CASE).....	23
1.9.1. Visual Paradigm.....	23
1.10. Metodología de desarrollo	24
1.10.1. OpenUP	24
1.11. Conclusiones parciales	25
CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DE LA APLICACIÓN	26
2.1. Propuesta de solución.....	26
2.2. Especificación de Requisitos de Software	29
2.2.1. Requisitos funcionales	30
2.2.2. Requisitos no funcionales.....	31
2.2.3. Diagrama de Casos de Uso del Sistema	33
2.2.4. Especificación de Casos de Uso.....	34
2.3. Descripción de estilos arquitectónicos y patrones de diseño	37

2.3.1. Patrones arquitectónicos	37
2.3.2. Patrones de diseño.....	38
2.4. Diagrama de Clases del Diseño	42
2.4.1. Diagramas de interacción.....	46
2.5. Diseño de la base de datos.....	47
2.5.1. Modelo Entidad-Relación de la base de datos.....	47
2.6. Modelo de Despliegue.....	49
2.7. Conclusiones parciales	50
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA APLICACIÓN.....	51
3.1. Modelo de componentes.....	51
3.2. Código fuente	53
3.3. Pantallas principales de la aplicación	53
3.4. Validación del sistema.....	55
3.4.1. Pruebas funcionales.....	55
3.4.2. Pruebas de seguridad	58
3.4.3. Pruebas de carga y estrés	60
3.4.4. Pruebas de usabilidad.....	60
3.5. Conclusiones parciales	62
CONCLUSIONES	64
RECOMENDACIONES.....	65
REFERENCIAS BIBLIOGRÁFICAS	66
BIBLIOGRAFÍA	69
ANEXOS	71
Anexo 1. Diagramas de proceso.....	71
Anexo 2. Descripción de Casos de Uso.....	76
Anexo 3. Estándares de codificación.....	80
Anexo 4. Pantallas principales de la aplicación	83
Anexo 5. Diseño de casos de prueba basados en casos de uso	85

ÍNDICE DE FIGURAS

Figura 1. Ciclo de vida de un audio o vídeo en el sistema.....	27
Figura 2. Diagrama de Casos de Uso del sistema.	34
Figura 3. Esquema simplificado de la arquitectura interna de Symfony. (34)	37
Figura 4. Diagrama de clases del diseño del caso de uso Autenticar Usuario.....	43
Figura 5. Diagrama de secuencia del caso de uso Autenticar Usuario.....	46
Figura 6. Modelo Entidad - Relación.	48
Figura 7. Diagrama de despliegue.	49
Figura 8. Diagrama de componentes de TeVeo V1.0.....	52
Figura 9. Vista de la aplicación para el usuario colaborador.	54
Figura 10. Comportamiento de las no conformidades en las pruebas funcionales.	58
Figura 11. Comportamiento de los indicadores de usabilidad en el SGDMP.....	62
Figura 12. Comportamiento de los indicadores de usabilidad en TeVeo Plus V1.0.....	62
Figura 13. Proceso identificación.	71
Figura 14. Proceso Selección.	72
Figura 15. Proceso Editorial.....	73
Figura 16. Proceso Documental.....	74
Figura 17. Proceso Almacenamiento.	75
Figura 18. Vista inicial de la aplicación.....	83
Figura 19. Vista de administración.....	84

ÍNDICE DE TABLAS

Tabla 1. Especificación de requisitos CU Autenticar Usuario.	36
Tabla 2. Clase: AlfrescoSetting.	44
Tabla 3. Clase: Alfresco.	45
Tabla 4. Clase: CE_Conectados.	45
Tabla 5. Diseño de caso de prueba del caso de uso Autenticar Usuario.	56
Tabla 6. Variables empleadas en el diseño de caso de prueba Autenticar Usuario.	57
Tabla 7. Comportamiento de las no conformidades por iteraciones.	57
Tabla 8. Estadísticas obtenidas a partir de la lista de chequeo de usabilidad.	61
Tabla 9. Especificación de requisitos CU Adicionar Contenidos.	77
Tabla 10. Especificación de requisitos CU Editar Contenidos.	78
Tabla 11. Especificación de requisitos CU Eliminar Contenidos.	79
Tabla 12. Diseño de caso de prueba del caso de uso Adicionar contenido.	86
Tabla 13. Variables empleadas en el diseño de caso de prueba Adicionar contenido.	86
Tabla 14. Diseño de caso de prueba del caso de uso Eliminar contenido.	86

INTRODUCCIÓN

La era digital o electrónica de la información comienza a gestarse en la década de los años cincuenta del siglo XIX. A partir de ello, las nuevas tecnologías son aplicadas en los distintos ámbitos del quehacer cotidiano del hombre, al punto que han revolucionado la sociedad de una manera sorprendente y su desarrollo ha contribuido a la modernización y a un manejo totalmente diferente de la información.

Desde los años sesenta del pasado siglo los rápidos avances en las tecnologías de la información han alcanzado a disímiles esferas, proponiendo formas óptimas de realizar tareas y automatizando procesos como en el caso de los Archivos¹. En este campo se comenzó a utilizar la informática para gestionar los grandes volúmenes de información que se generaban, dando paso al surgimiento de soluciones y servicios de gestión documental digitalizados.

La gestión documental es una función clave que debe estar alineada con la estrategia y los objetivos de negocio, orientada a los procesos e integrada con los sistemas de la información y gestión del conocimiento, que dan soporte a estos procesos. Es un conjunto de normas, técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización, permite la recuperación de la información desde ellos, determina el tiempo que los documentos deben guardarse, eliminar los que ya no sirven y asegurar la conservación indefinida de los más valiosos, aplicando principios de racionalización y economía. (1)

A principios de los años ochenta fue desarrollado el término mediateca cuando los contenidos audiovisuales (documentos sonoros y grabaciones de vídeo) fueron considerados como testimonios culturales al igual que los materiales escritos. El término mediateca ha sido acuñado para reflejar mejor la diversidad de las obras y recursos reunidos y presentados bajo la forma de videocintas en formatos *Betacam*² y VHS³. En los años noventa, las mediatecas comenzaron a almacenar soportes numéricos (CD⁴ para audio y DVD⁵ para vídeo) que vinieron a completar los soportes tradicionales (impresos, microfilmes, discos de vinilo, *cassettes*).

Cada día el volumen e importancia de los recursos multimedia almacenados en las unidades de

¹ Conjunto ordenado de documentos que una persona, una sociedad o institución producen el ejercicio de sus funciones o actividades.

² Familia de formatos de videocintas profesionales de media pulgada (1/2") creada por Sony en 1982.

³ Siglas del Inglés *Video Home System*, sistema de grabación y reproducción analógica de audio y vídeo.

⁴ Siglas del Inglés *Compact Disc* (Disco Compacto).

⁵ Siglas del Inglés *Digital Versatile Disc* (Disco Versátil Digital), disco de almacenamiento de audio, vídeo y datos.

información⁶ ascienden como resultado de la constante demanda que han adquirido. Las unidades atesoran y conservan una enorme cantidad de información que puede ser transformada en contenidos accesibles a través de la Web⁷ para diferentes tipos de usuarios. Dicha transformación impone la creación de estructuras de información y aplicaciones informáticas capaces de automatizar los procesos archivísticos tradicionales y que fomenten la difusión de las mejores prácticas en los sistemas internos de trabajo, es por ello que han aparecido soluciones como los Gestores de Contenido Empresarial (ECM).

Un ECM reúne las estrategias, métodos y herramientas utilizadas para capturar, gestionar, almacenar, preservar y entregar contenidos y documentos relacionados con los procesos organizativos (2). En general es un término que abarca la gestión de documentos, gestión de contenidos web, colaboración, gestión de registros, gestión de multimedias, gestión del flujo de trabajo, búsqueda y digitalización.

En Cuba, a pesar del auge de las tecnologías, aún existen unidades de información que carecen de un sistema que les permita realizar la gestión y documentación de los archivos de audio y vídeos, propiciando que la información quede expuesta a riesgos de pérdida o uso indebido, no solo por robo o extravío sino también por rotación de personal, fallas en equipos o el simple desorden, a veces incontrolable, provocado por el crecimiento continuo de los medios de comunicación. (3)

Tomando en cuenta que por la falta de una gestión adecuada, el patrimonio de audio y vídeos de las instituciones no es accesible al público que lo necesita, y que la falta de normalización y estandarización en la representación y transmisión de los recursos de audio y vídeos provoca la desvalorización de los recursos multimedia, en la Universidad de las Ciencias Informáticas (UCI) se desarrolló el Sistema de Gestión Documental de Medias para la Prensa haciendo uso del ECM Alfresco. (4)

Este sistema permite la gestión y documentación de los archivos de audio y vídeos haciendo uso de la interfaz predeterminada de Alfresco (Ver sección 1.3 Alfresco Explorer), pero su arquitectura no es la adecuada: tiende a incrementarse la complejidad del sistema y a disminuir su flexibilidad haciendo que la experiencia del usuario no sea óptima y en ocasiones, frustrante. Entre la serie de inconvenientes identificados se destacan los siguientes:

- El diseño poco intuitivo de la interfaz gráfica de usuario de dicha aplicación web no resulta atractivo, lo cual provoca una alta desmotivación por parte del usuario cuando trabaja con el sistema.

⁶ Cualquier tipo de biblioteca, ya sean las nacionales, públicas, universitarias o especializadas. Asimismo, los diferentes archivos, tanto los administrativos como los históricos, pueden ser considerados unidades de información.

⁷ Siglas del Inglés World Wide Web, (o La Web) el sistema de documentos (o webs) interconectados por enlaces de hipertexto, que se ejecutan en Internet.

- El sistema automatiza una serie de procesos relacionados con la gestión documental de archivos de audio y vídeos que se realizan completamente tras ejecutar una larga lista de pasos, ocasionando que el usuario no tenga idea de qué opciones seleccionar.
- Se hace lenta la gestión y recuperación de materiales multimedia debido a que el usuario debe realizar varios procesos que pueden ser optimizados.
- La visibilidad del sistema se ve afectada por la existencia de información oculta que necesita acciones específicas para ser visualizadas, además los enlaces del menú apenas resaltan cuando se seleccionan.
- El sistema no se adecúa a las necesidades de los usuarios pues no tiene forma de reportar fallos detectados o sugerir cambios, el lenguaje empleado en los menús está más cercano al informático o técnico que al utilizado por el usuario, no existe una manera obvia de acceder a páginas relacionadas con otras secciones.
- La estructura del sistema no resulta evidente para el usuario, la manera de navegar por la Web o ejecución de tareas asignadas requieren de práctica y conocimientos web, los íconos asociados con contenido no se relacionan fácilmente.
- La flexibilidad y eficiencia de uso se ven afectadas por tareas repetitivas que no se efectúan con facilidad, no se evitan los pasos inútiles, las partes o secciones más importantes del sistema no son accesibles desde la página de inicio y no existen accesos rápidos a operaciones frecuentes.

A partir de lo expresado anteriormente se ha determinado como **problema de la investigación**: ¿Cómo mejorar la gestión de audio y vídeos en el Sistema de Gestión Documental de Medias para la Prensa?

El **objeto de estudio** se centra en el desarrollo de aplicaciones web para la administración de contenido multimedia.

El **campo de acción** se basa en el desarrollo de aplicaciones web para los Sistemas de Gestión Documental de Audio y Vídeos Digitales.

Se propone como **objetivo general** desarrollar una aplicación web que mejore la gestión de audio y vídeos en el Sistema de Gestión Documental de Medias para la Prensa.

Para complementar el objetivo de la investigación se han formulado los siguientes **objetivos específicos** que posibilitan el desarrollo del mismo:

- Evaluar las tendencias de las aplicaciones web para los sistemas de gestión documental.
- Diseñar las funcionalidades de la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0.
- Implementar las funcionalidades de la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0.
- Validar las funcionalidades de la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0.

Teniendo como premisa lograr dichos objetivos se plantearon las siguientes tareas de investigación:

- Caracterización de las tendencias de diseño e implementación de las aplicaciones web para sistemas de gestión documental.
- Selección de las tecnologías, herramientas, estándares, patrones y metodologías necesarias durante el desarrollo de la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0.
- Identificación de los requerimientos funcionales y no funcionales de la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0.
- Elaboración de la arquitectura de información de la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0.
- Diseño de la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0.
- Implementación de las funcionalidades de la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0.
- Ejecución de las pruebas de funcionalidad, seguridad, carga y estrés y usabilidad a la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0.

Durante el desarrollo de la presente investigación se hará uso de métodos científicos para dar cumplimiento a los objetivos planteados, entre ellos se identifican los siguientes:

Métodos Teóricos

- **Analítico-Sintético:** para realizar el análisis y la síntesis de los conceptos, estándares y herramientas en la investigación sobre las mediatecas. Fue usado en la revisión de documentos y artículos, de donde se extrajeron las ideas centrales relacionadas con el funcionamiento de las

aplicaciones web para la gestión documental. Permitted obtener el conocimiento esencial para generar una propuesta de solución adecuada a las exigencias de las unidades de información.

- **Histórico-Lógico:** usado en la revisión de las tendencias de aplicaciones desarrolladas para la gestión documental con el Alfresco. Permitted documentar los elementos de juicio necesarios para garantizar la comprensión y un mejor funcionamiento de estas aplicaciones.

Con el desarrollo de la investigación se espera obtener una aplicación web para la gestión de los contenidos en el Sistema de Gestión Documental de Audio y Vídeos Digitales, que permita identificar, describir, catalogar, recuperar, publicar y compartir los recursos multimedia, además de generar automáticamente diferentes perfiles a partir del archivo de audio o vídeo adicionado originalmente al sistema.

El presente trabajo está conformado de la siguiente manera: introducción, tres capítulos que serán descritos a continuación, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos.

Capítulo 1: Fundamentación teórica. Se hace referencia a los principales conceptos relacionados con la gestión de archivos digitales de audio y vídeo, y se caracterizan las tendencias actuales de las aplicaciones web para los sistemas de gestión documental. Asimismo, se hace una investigación de las diferentes tecnologías, herramientas y metodologías que se utilizarán en el desarrollo del presente trabajo.

Capítulo 2: Características y diseño de la aplicación. Es caracterizada la solución propuesta, describiendo cada uno de los procesos que intervienen a través de diagramas; son identificados y descritos los requisitos funcionales y no funcionales del sistema. También se describen los principales estilos arquitectónicos y patrones de diseño utilizados en la solución. De igual modo se proponen los diagramas de clases del diseño del sistema, diagramas de interacción y el modelo de despliegue.

Capítulo 3: Implementación y validación de la aplicación. Para desarrollar la solución propuesta se realiza todo el proceso de implementación del software. Son representados los diagramas de componentes y las implementaciones relevantes del código fuente. Finalmente se procede a la validación del sistema a través de la realización de pruebas de funcionalidad, seguridad, carga y estrés y usabilidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La gestión documental de archivos de audio y vídeos digitales, surge como una necesidad de la sociedad contemporánea. Una solución relacionada con esta área involucra una serie de conceptos capaces de mejorar su comprensión, además de un estudio lo suficientemente profundo como para reflejar las deficiencias y necesidades más actuales. Así mismo el desarrollo de software es todo un proceso que lleva implícito un conjunto de herramientas y tecnologías, que según las exigencias del cliente, requieren de una selección minuciosa que tenga en cuenta los principios de seguridad, eficiencia y rendimiento básicos de la industria informática.

1.1. Revisión conceptual de la propuesta de solución

Aplicable a todo tipo de documento administrativo o de archivo, la gestión documental constituye una rama muy amplia. Su terminología más actual abarca la gestión documental de archivos de audio y vídeos digitales, surgiendo como una necesidad de la sociedad actual. Este término es empleado en diversas fuentes de información con un enfoque a determinadas áreas por lo que para su mejor entendimiento se hace necesario referenciar una serie de conceptos que se relacionan con la expresión: Gestión documental de archivos de audio y vídeos digitales.

1.1.1 Gestión documental

La gestión documental es un área de la administración general que se encarga de garantizar la economía y eficiencia en la creación, mantenimiento, uso y disposición de los documentos administrativos durante todo su ciclo de vida. (5)

De cierta forma, la gestión de documentos incluye una serie de operaciones que influyen en la economía y en la eficiencia de las áreas de producción, mantenimiento y uso de los documentos durante su ciclo de vida. Todo sistema de gestión de documentos adoptado por cualquier organización persigue facilitar el trabajo, evitar que la información se duplique, y conservar el patrimonio de la institución.

La gestión documental es un proceso administrativo que permite analizar y controlar sistemáticamente, a lo largo de su ciclo de vida, la información registrada que crea, recibe, mantiene o utiliza la organización en correspondencia con su misión, objetivos y operaciones; se considera un proceso para mantener la información en un formato que permita su acceso oportuno, y por ello se requiere de la realización de tareas y procedimientos particulares para cada fase de su ciclo de vida y su explotación. La información

registrada es evidencia de las actividades y transacciones de las organizaciones, y su uso oportuno permite a la organización una mayor eficacia en su labor. (6)

Uno de los aspectos más importantes de la gestión documental desde el enfoque de los Archivos totales, es la implantación de sistemas integrales de tratamiento de la documentación que se ocupen de su producción, conservación, uso, explotación y eliminación, según las necesidades de cada tipo de usuario, sean los propios productores o investigadores. (7)

Luego de un análisis de cada definición, se entiende que la gestión documental es todo un proceso que involucra al patrimonio documental de una organización, con el objetivo de preservar la información durante su ciclo de vida y tenerla disponible en el formato y momento deseados para facilitar su manipulación, logrando así mayor eficiencia en el desarrollo de las tareas de la institución.

1.1.2 Mediatecas

El origen de las mediatecas digitales se encuentra históricamente ligado a la asimilación e incorporación por parte de las bibliotecas del desarrollo informático, pues estas se fueron abriendo al paso de la informática; incorporando y adaptando en sí mismas espacios dedicados a las computadoras, con el objetivo de que fueran utilizadas para consultas de información. Inicialmente se conocía como mediateca el espacio dedicado dentro de las bibliotecas a realizar consultas a la información digital y búsquedas en Internet. En la actualidad, la mediateca digital hace suyo un espacio mucho más amplio, pues es utilizada por las empresas, las instituciones y otros organismos para poner a disposición de todos de modo digital la información que dichas entidades necesiten manipular o consultar.

Mediateca digital es un centro de consulta de información digital, de exploración de medios de comunicación y de experimentación de nuevas formas de arte y expresión ligadas con la informática. Nuevos materiales como discos compactos (CD-ROM⁸ y DVD), microfichas⁹, audio *cassettes* y filmaciones (vídeos, microfilmes¹⁰); permiten almacenar grandes cantidades de información, incluso estos pueden transitar por la mediateca a través de bases de datos en línea, sin estar alojados en su interior, físicamente sólo se requieren las terminales de consulta y del sitio o área de máquinas. (8)

1.1.3 Audio digital

Proveniente del Inglés *audio*, y del Latín *audīo*, yo oigo. Un archivo de audio se refiere a la técnica relacionada con la reproducción, grabación y transmisión del sonido. (9)

⁸ Siglas del inglés *Compact Disc-Read Only Memory* es un disco compacto prensado que contiene los datos de acceso, pero sin permisos de escritura, un equipo de almacenamiento y reproducción de música

⁹ Ficha de película que contiene en tamaño muy reducido varias imágenes o fotografías de un libro o documento.

¹⁰ Sistema de archivo, gestión y difusión documental.

En la rama de la electrónica el audio digital se considera como la codificación digital de una señal eléctrica que representa una onda sonora. Consiste en una secuencia de valores enteros y se obtienen de dos procesos: el muestreo y la cuantificación digital de la señal eléctrica.

Los archivos de audio digital almacenan toda la información que ocurra en el tiempo. El tamaño del archivo no varía, así contenga silencio o sonidos muy complejos. Existen muchos formatos de archivo de audio digital, los cuales pueden clasificarse en dos categorías PCM¹¹ (incluye los formatos WAV¹², AIFF¹³ y RAW¹⁴) y comprimidos (incluye los formatos MP3¹⁵, AAC¹⁶ y Ogg¹⁷).

Finalmente, los autores de esta investigación puntualizan que en el sistema de gestión documental de archivos digitales: TeVeo Plus V1.0 el audio será tratado como un archivo de datos provisto de sonido.

1.1.4. Vídeo digital

La Proveniente del Inglés *video*, y del Latín *vidēo*, yo veo. Un archivo de vídeo se refiere al sistema de grabación y reproducción de imágenes, acompañadas o no de sonidos, mediante cinta magnética. (10)

En la rama de la electrónica el vídeo digital es un tipo de sistema de grabación de vídeo que funciona usando una representación digital de la señal, en vez de una señal analógica. Actualmente, el término hace referencia a distintos formatos. Además de las cintas de vídeo analógico, como VHS¹⁸ y *Betamax*¹⁹, también se incluyen los formatos digitales DVD²⁰ y MPEG4²¹. La calidad del vídeo estará determinada por distintos factores, como el método de captura y el tipo de almacenamiento elegido.

Finalmente, los autores de esta investigación puntualizan que en el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0 estos recursos serán tratados como archivos de datos que contienen una sucesión de imágenes reales convertidas a lenguaje binario, que a su vez están provistos de sonido.

1.1.5. Gestores de Contenidos Empresariales (ECM)

Los ECM son aplicaciones que se están imponiendo en las organizaciones. Actualmente ECM es un

¹¹ Modulación por Codificación de Pulsos. Técnica de muestreo para digitalizar señales análogas.

¹² WAV (o WAVE), apócope de *WAVEform audio file format*, es un formato de audio digital normalmente sin compresión de datos.

¹³ Formato de Intercambio de Archivos de Audio.

¹⁴ Formato de Imagen sin modificaciones es un formato de archivo digital de imágenes que contiene la totalidad de los datos de la imagen tal y como ha sido captada por el sensor digital de la cámara fotográfica.

¹⁵ Formato de compresión de audio digital.

¹⁶ Codificación de Audio Avanzada.

¹⁷ Formato contenedor para los códec multimedia.

¹⁸ Siglas del Inglés *Video Home System*, sistema de grabación y reproducción analógica de audio y vídeo.

¹⁹ Formato de vídeo analógico.

²⁰ Disco Versátil/Vídeo Digital. Formato de almacenamiento digital de datos.

²¹ Formato para la compresión digital de audio y vídeo.

término genérico que engloba soluciones de gestión documental, gestión de contenidos web, colaboración, gestión de registros, gestión de activos digitales DAM²², gestión de flujos de trabajo, captura y digitalización de elementos integrado con firmas electrónicas de larga duración.

El objetivo principal de un ECM es facilitar la gestión corporativa en cuanto al almacenamiento, seguridad, control de versiones y flujos de trabajo. En resumen, los ECM mejoran la eficiencia mediante la reducción de costes y mejora en el control. (11)

1.1.6. Características del ECM Alfresco

Alfresco es el líder en el mercado de código abierto para la gestión de contenidos empresariales. Cumple con la filosofía de software libre y está desarrollado en el lenguaje de programación orientado a objetos Java. Ofrece una arquitectura flexible para proveer la gestión de documentos.

Además, puede estar en muchas máquinas con distintas configuraciones funcionando como uno solo y prestar servicios para muchos o pocos usuarios. Presenta un alto nivel de seguridad a través de la gestión de usuarios, grupos y roles.

Las aplicaciones construidas usando Alfresco pueden ser desplegadas en cualquier ambiente. Permite usar gestores de base de datos como PostgreSQL²³, MySQL²⁴, entre otras; e incluso puede ejecutarse sobre cualquier servidor de aplicaciones que soporte Java y trabajar con el navegador web que desee.

Ofrece diferentes herramientas y funciones, entre las que se puede destacar la gestión de audio y vídeos digitales, brinda un repositorio integrado para manejar todos los contenidos. Además, posibilita la extracción de los metadatos y la captura, etiquetamiento y enrutamiento de recursos multimedia. También brinda una amplia gama de paquetes y librerías.

1.1.7. Usabilidad

La usabilidad (dentro del desarrollo web) es la disciplina que estudia la forma de diseñar sitios web para que los usuarios puedan interactuar con ellos de la forma más fácil, cómoda e intuitiva posible. (12)

Numerosos autores han propuesto diversas definiciones de usabilidad, normalmente a través de la enumeración de los diferentes atributos o factores mediante los que puede ser evaluada, cada definición depende del enfoque con el que pretende ser medida. (13)

²² Siglas del Inglés *Digital Asset Management* (Gestión de Recursos Digitales).

²³ Sistema Gestor de Bases de Datos.

²⁴ Sistema Gestor de Bases de Datos.

Para este trabajo se utiliza la definición ofrecida por la ISO²⁵ y que define usabilidad como el grado de eficacia, eficiencia y satisfacción con la que usuarios específicos pueden lograr objetivos específicos, en contextos de uso específico. (14)

Actualmente la mayoría de los sitios web están centrados en los usuarios, para facilitarles la adquisición de la información deseada, mediante el uso de imágenes, multimedia y otros materiales de apoyo.

Dentro de los aspectos contenidos en la usabilidad se encuentran: la utilidad del producto (no utilidad funcional), la facilidad de aprendizaje, la memorización, eficacia, eficiencia (errores) y la satisfacción del usuario final. Según el perfil, la importancia de la usabilidad radica en que para el usuario: es la diferencia entre hacer una tarea de forma precisa y completa o no, y que el proceso sea ameno o frustrante; para el desarrollador: marca la diferencia entre el éxito o el fallo de un sistema.

1.1.8. Accesibilidad

Un concepto íntimamente ligado al de usabilidad es el de accesibilidad. Éste ya no se refiere a la facilidad de uso, sino a la posibilidad de acceso. En concreto a que el diseño, como prerequisite imprescindible para ser usable, posibilite el acceso a todos sus potenciales usuarios, sin excluir a aquellos con limitaciones individuales: discapacidades, dominio del idioma, o limitaciones derivadas del contexto de acceso: software y hardware empleado para acceder, ancho de banda de la conexión empleada, etc. (15)

La accesibilidad web es la posibilidad que presenta un producto o servicio web de ser accedido y usado por el mayor número posible de personas, indistintamente de las limitaciones propias del individuo o de las derivadas del contexto de uso. (16)

Un diseño será accesible cuando sea usable para más personas en más situaciones o contextos de uso (17), lo que posibilita a todos los usuarios, de forma eficiente y satisfactoria, la realización y consecución de tareas. (18)

1.2. Tendencias actuales de las interfaces web desarrolladas para el manejo de archivos multimedia

Para determinar las condiciones en que se encuentra actualmente la gestión documental de audio y vídeos digitales se caracterizaron un conjunto de aplicaciones existentes. Dentro de estas se incluyen a nivel internacional YouTube y Vimeo, en el ámbito nacional: teVeo y en el marco de la UCI: Primicia.

²⁵ Organización Internacional para la Estandarización, que regula una serie de normas para fabricación, comercio y comunicación, en todas las ramas industriales.

1.2.1. YouTube

YouTube es una red social que se ofrece como plataforma de distribución para creadores de contenido y publicistas originales. Permite cargar, buscar y ver contenido de vídeo. (19)

Características de YouTube:

- ✓ Maneja varios tipos de contenido entre los que se encuentran textos escritos, audio, vídeo e imágenes fijas.
- ✓ Permite subir y editar vídeos, realizar búsquedas y crear listas de reproducción.
- ✓ Mantiene una logística que permite localizar cualquier vídeo por medio de las etiquetas de metadatos, títulos y descripciones que los usuarios asignan a sus vídeos.
- ✓ Las API²⁶ del reproductor permiten controlar la reproducción de vídeos de YouTube en otros sitios web, así como configurar opciones básicas, controlar la interfaz del reproductor o, incluso, crear controles propios del reproductor.
- ✓ Brinda una interacción diferida y en tiempo real (chat), donde el usuario puede enviar comentarios, subir vídeos y hacer otras operaciones.
- ✓ Permite compartir vídeos vía correo electrónico, redes sociales o blog, directamente desde la página de vídeos.
- ✓ Cuenta con una interfaz ligera, información concisa y varias opciones de edición en pantalla.

1.2.2. Vimeo

Vimeo es una red social de Internet basada en vídeo. Vimeo no admite comerciales de televisión, demostraciones de videojuegos, pornografía o cualquier contenido que no haya sido creado por el usuario. El sitio, además, ha ganado reputación como proveedor de imágenes para diversos artistas, debido a la alta tasa de bits²⁷ y resolución de sus vídeos. (20)

Características de Vimeo:

- ✓ Entre los tipos de contenido que maneja se encuentran audio, vídeo e imágenes fijas.
- ✓ Permite almacenar vídeos digitales para que los usuarios comenten en sus páginas.
- ✓ Permite compartir vídeos ofreciendo controles de privacidad para decidir quiénes podrán ver el

²⁶ Interfaz de Programación de Aplicaciones, es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.

²⁷ Unidad de medida de la capacidad de memoria de un ordenador, o de un disco magnético.

vídeo y dónde.

- ✓ Los usuarios deben estar registrados para subir vídeos, crear su perfil, cargar avatares, comentar y armar listas de favoritos.
- ✓ Permite la creación de canales, grupos y álbumes sin restricción, así como la visualización de vídeos libres de publicidad.

1.2.3. teVeo

teVeo es una aplicación web que ofrece el servicio de audio y vídeo bajo demanda del Instituto Cubano de Radio y Televisión (ICRT) (21), es decir, permite al usuario el acceso a contenidos multimedia de forma personalizada, dando la posibilidad de solicitar y visualizar los contenidos en el momento deseado.

Características de teVeo:

- ✓ El contenido disponible en el sitio incluye los tipos: audio y vídeo.
- ✓ Los archivos son organizados según su contenido, actualización y popularidad en bloques como: Recomendaciones, Lo más reciente y Lo más popular en teVeo.
- ✓ Permite realizar búsqueda simple mostrando la información de los archivos de audio y vídeos, cuenta además con un servicio RSS, para publicar las últimas actualizaciones del sitio.
- ✓ Cuenta con permisos limitados a la institución, sin opciones para establecer contacto con los administradores, comentar, o compartir archivos.

1.2.4. Primicia

Primicia es una aplicación informática capaz de visualizar informaciones de texto a través de la TV, tomando como base fundamental las características básicas de otros sistemas e incluye algunas variaciones importantes como es la transmisión de materiales audiovisuales: imágenes y vídeos.

Características de Primicia:

- ✓ Posee funcionalidades genéricas fácilmente escalables, se orienta a proveer un canal de televisión básicamente con características informativas, donde se integran contenidos de texto, imagen fija y vídeo.
- ✓ Incluye los elementos necesarios para la transmisión de un canal de televisión, como son imagen gráfica, configuración de la transmisión, gestión de contenidos, entre otros.
- ✓ Se encuentra estructurado en dos subsistemas que se relacionan entre sí. El Subsistema de

Administración es el responsable de la administración del canal y toda la gestión de las noticias y recursos multimedia. El Subsistema de Transmisión es el encargado de visualizar las noticias y materiales publicados.

Primicia integra elementos que presentan soluciones más grandes y costosas, logrando hacerlo de una forma más compacta y sencilla que permite incluirse como una buena opción para canales de televisión con varios entornos. (22)

1.2.5. Observaciones

Luego de un análisis de las características de las aplicaciones web desarrolladas para la gestión de archivos multimedia se determinó que la mayoría de ellas:

- ✓ Hacen uso de las tecnologías más avanzadas en cuanto a desarrollo web.
- ✓ Se benefician del valor agregado que ofrece la integración con herramientas interactivas diferidas y en tiempo real.
- ✓ Se caracterizan por la simplicidad para interactuar con los contenidos multimedia.

En cuanto a la observación de algunos elementos de usabilidad, todas estas aplicaciones:

- ✓ Poseen una URL²⁸ correcta, clara y fácil de recordar.
- ✓ La terminología del menú es constante.
- ✓ Los enlaces de menú resaltan cuando se seleccionan.
- ✓ Los íconos se identifican con lo que representan.
- ✓ No es necesario que el usuario tenga experiencia para navegar por el sitio.
- ✓ Tiene formas para contactar con los administradores de la aplicación, excepto TeVeo.
- ✓ Muestran mensajes de errores claros y entendibles.
- ✓ Funcionan sobre los navegadores más usados, entiéndase Internet Explorer, Chromium y Mozilla Firefox.
- ✓ No contienen enlaces rotos o que no lleven a ningún sitio.
- ✓ El acceso a la ayuda está en una zona visible y reconocible.

Sin embargo presentan los siguientes inconvenientes:

²⁸ Siglas del Inglés de *Uniform Resource Locator* (Localizador de Recursos Uniforme) es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.

- ✓ Los archivos de audio y vídeos son almacenados sin tener en cuenta normas ni estándares para su representación, conservación y catalogación.
- ✓ Sus objetivos no son facilitar el trabajo, ni evitar que la información se duplique, y mucho menos conservar el patrimonio documental de instituciones.
- ✓ No dan seguimiento a los archivos de audio y vídeos durante su ciclo de vida como material documental o archivístico de una institución.

1.3. Tendencias actuales de las interfaces web desarrolladas para el ECM Alfresco

Teniendo en cuenta que la aplicación que se pretende desarrollar hace uso del ECM Alfresco y requiere de una nueva interfaz gráfica de usuario se caracterizaron sistemas que han implementado soluciones similares. Entre estos están Alfresco Explorer, Opsoro, iFresco a nivel internacional, así como eXcriba dentro de la UCI.

1.3.1. Alfresco Explorer

Alfresco Explorer es la aplicación web clásica del ECM Alfresco. Es su principal aplicación para la gestión de documentos, proporciona una completa interfaz web de usuario con capacidades de gestión de documentos, colaboración y administración (23). Actualmente el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeó Plus V1.0 hace uso de esta interfaz.

Características de Alfresco Explorer:

- ✓ Es de código abierto, liberado bajo la licencia LGPL²⁹.
- ✓ Consume los servicios directamente del repositorio del ECM Alfresco.
- ✓ Está desarrollado en Java.
- ✓ Se basa en la tecnología JavaServer Faces³⁰, utilizando la implementación de Apache MyFaces³¹.

Sin embargo entre sus limitaciones se destacan las siguientes:

- ✓ Su diseño está orientado a usuarios con conocimientos avanzados en las tecnologías informáticas.
- ✓ Los servicios que brinda no incluyen los necesarios para la gestión de los archivos multimedia en su ciclo de vida.

²⁹ Lesser GPL (LGPL) o GPL reducida de GNU.

³⁰ Tecnología y *framework* para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.

³¹ Proyecto de *Apache Software Foundation*, para mantener una implementación abierta de *JavaServer Faces* JSF.

- ✓ Su configuración es muy complicada.
- ✓ La información de consulta disponible es escasa.

1.3.2. Opsoro

Opsoro es un cliente web alternativo para el ECM Alfresco. Su objetivo es proporcionar una interfaz sencilla y fácil de manejar por el usuario, mediante el aprovechamiento de las últimas tecnologías web.

Características de Opsoro:

- ✓ Es un software de código abierto publicado bajo la licencia GPL³².
- ✓ Está desarrollado en la librería ExtJS para Ajax³³.
- ✓ Utiliza la tecnología Web Script.
- ✓ Es amigable e intuitivo para el usuario.
- ✓ Permite la búsqueda a medida que se escribe.
- ✓ Contiene categorías y etiquetas de carpetas.
- ✓ Permite la vista previa de archivos.
- ✓ Opsoro utiliza principalmente JSON³⁴ como formato de respuesta para mantener los fragmentos de datos tan pequeños como sea posible.
- ✓ Proporciona una interfaz de usuario sencilla. (24)

A pesar de sus ventajas esta interfaz web no tiene las funcionalidades requeridas para realizar una gestión de archivos multimedia durante todo su ciclo de vida.

1.3.3. iFresco

iFresco es un cliente web de código abierto desarrollado para el ECM Alfresco y publicado bajo la licencia GPL, que puede ser manejado con facilidad. Este sistema es muy sencillo de utilizar, pues posee funcionalidades como menús contextuales, listas ordenadas y permite arrastrar y soltar elementos. (25)

Características de iFresco:

- ✓ Se implementó en PHP³⁵.

³² Licencia Pública General.

³³ Del Inglés *Asynchronous JavaScript And XML*, es una técnica de desarrollo web para crear aplicaciones interactivas.

³⁴ Formato ligero para el intercambio de datos.

³⁵ Siglas del Inglés *Hypertext Preprocessor*, lenguaje de programación del lado del servidor.

- ✓ Está desarrollado en Symfony 1.4.
- ✓ Utiliza el ORM³⁶ Doctrine³⁷ para el mapeo a la base de datos.
- ✓ Como Sistema Gestor de Base de Datos utiliza MySQL para salvar datos administrativos, las sesiones de usuarios y favoritos.
- ✓ Utiliza jQuery para diversas funciones de JavaScript.
- ✓ La integración con el ECM Alfresco está basada en servicios web API.

Como limitaciones se identificaron:

- ✓ No posee las funcionalidades y características necesarias para una correcta gestión de archivos multimedia en todo su ciclo de vida.
- ✓ No existe documentación sobre el desarrollo de la tecnología.
- ✓ No cuenta con una comunidad capacitada para su seguimiento y corrección de las deficiencias y errores de seguridad identificados.

1.3.4. eXcriba

eXcriba es un sistema creado para la gestión de grandes cantidades de documentos. Es desarrollado por ingenieros del Centro de Informatización Universitaria de la Facultad 1 de la Universidad de las Ciencias Informáticas.

Características de eXcriba:

- ✓ Guía y ayuda al usuario a realizar una gestión de los documentos de una forma adecuada, sencilla y organizada. (26)
- ✓ Es un sistema amigable, de fácil uso, orientado a usuarios de poco conocimiento de la informática.

Desventajas de eXcriba:

- ✓ La versión del ECM Alfresco para el cual está desarrollado no es compatible con las especificaciones del sistema.
- ✓ No tiene en cuenta la especialización de los tipos de contenidos, dado que trata todos los documentos de igual forma.

³⁶ Siglas del Inglés *Object-Relational Mapping* (Mapeo Objeto-Relacional), técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

³⁷ Biblioteca integrada con Symfony, para facilitar la persistencia y lectura de información hacia y desde una base de datos.

- ✓ No es flexible al cambio de las interfaces de usuarios, pues todos los usuarios tienen una misma interfaz.
- ✓ No brinda tratamiento a los archivos de audio y vídeos.

1.3.5. Observaciones

Tras la investigación relacionada con las aplicaciones que implementan interfaces para el ECM Alfresco, se determinó que:

- ✓ Todos hacen uso de novedosas tecnologías.
- ✓ Dan tratamiento de manera general a varios tipos de contenido.
- ✓ Ofrecen prestaciones esenciales para la gestión documental.
- ✓ Implementan buenas prácticas en cuanto a los mecanismos de integración con el Alfresco.

Sin embargo presentan los siguientes inconvenientes:

- ✓ Las interfaces gráficas de usuario no son de fácil usabilidad y exigen que el usuario final posea un conocimiento medio o avanzado en informática.
- ✓ No simplifican la administración de documentos a través de su ciclo de vida (captura, creación, clasificación, protección, compartir, retención de archivo y destrucción).
- ✓ No cumplen con las exigencias ni las características del proyecto de Proveedor de Audio y Vídeo para Internet, además no satisfacen las necesidades y restricciones del cliente en cuanto a la representación y manejo de audio y vídeos digitales.

1.4. Marcos de trabajo (*framework*)

Un *framework* es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software es más fácilmente organizado y desarrollado. Puede incluir soporte para programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, que contribuyan a desarrollar y unir los diferentes componentes de un proyecto. Dicho de otro modo, es un producto que se utiliza como base para la programación avanzada de aplicaciones, que aporta funciones o código para realizar las tareas más básicas, puesto que en el propio *framework* ya hay implementaciones que están probadas, funcionan y no es necesario volver a programarlas. Para el desarrollo de este trabajo se propone la utilización de los *framework* que se relacionan a continuación.

1.4.1. Symfony 2.1.4

Symfony es un *framework* diseñado para optimizar el desarrollo de aplicaciones web. Entre sus características se encuentra la de separar la lógica de negocio, la lógica de servidor y la presentación de la aplicación web, así como proporcionar varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de un sistema web complejo; además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel; es compatible con varios gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. (27)

1.4.2. jQueryUI 1.9.2

jQuery es un *framework* para el lenguaje JavaScript, que implementa una serie de clases para programar sin tener en cuenta el navegador para el cual estará destinada la aplicación. Ofrece una infraestructura que facilita la creación de aplicaciones complejas del lado del cliente. jQuery ayuda a la creación de interfaces gráficas de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax. Este *framework* tiene licencia para su uso en cualquier tipo de plataforma. (28)

jQuery UI es una biblioteca de componentes para el *framework* jQuery que le añaden a este un conjunto de *plugins*³⁸, *widgets*³⁹ y efectos visuales para la creación de aplicaciones web.

1.4.3. Bootstrap 2.2.2

Bootstrap es un *framework* desarrollado para simplificar el proceso de creación de diseños web. Para ello ofrece una serie de plantillas CSS⁴⁰ y de ficheros JavaScript que permiten conseguir interfaces web que funcionen en los navegadores actuales; un diseño que pueda ser visualizado de forma correcta en distintos dispositivos y a distintas escalas y resoluciones; una mejor integración con las librerías usadas habitualmente, como por ejemplo jQuery; un diseño sólido basado en herramientas actuales y potentes, o estándares como CSS3/HTML5⁴¹. (29)

1.4.4. Justificación de la selección

Para los desarrolladores el empleo de *framework* agiliza y facilita el trabajo; mayormente son utilizados

³⁸ Módulos de hardware o software que añaden una característica o un servicio específico a un sistema más grande.

³⁹ Pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños utilizados para dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

⁴⁰ Siglas del Inglés *Cascade Style Sheet* (Hojas de Estilo en Cascada).

⁴¹ Siglas del Inglés *Hyper Text Markup Language* (Lenguaje de Marcado de Hipertexto).

con el objetivo de evitar la redundancia de código y fomentar la reutilización. La selección de los *framework* anteriores se sustenta en que:

- ✓ Symfony 2.1.4 ha sido solicitado por el usuario final, pues en algún momento tuvo referencias de que es fácil de instalar y configurar en la mayoría de plataformas, es independiente del sistema gestor de bases de datos, es sencillo y flexible, implementa la mayoría de las mejores prácticas y patrones de diseño para la web. Además está preparado para aplicaciones empresariales y es adaptable a las políticas y arquitecturas propias de cada empresa.
- ✓ Symfony 2.1.4, para los desarrolladores genera un código fácil de leer y permite un mantenimiento sencillo. Es fácil de extender lo que posibilita su integración con librerías desarrolladas por terceros.
- ✓ jQueryUI 1.9.2 permite la creación de interfaces gráficas de usuario con efectos dinámicos y visuales compatibles con los navegadores especificados por el cliente y mucho más atractivos para el usuario.
- ✓ Bootstrap 2.2.2 facilita el trabajo con el lenguaje CSS garantizando la compatibilidad entre dispositivos a distintas escalas y resoluciones.

1.5. Lenguajes para la Web

Los lenguajes para la Web son aquellos asimilados directamente por el navegador y que no necesitan pretratamiento. En la actualidad existe una gran diversidad de ellos, que han surgido como consecuencia de las tendencias y necesidades de las distintas plataformas de desarrollo. El uso de estos lenguajes permite un diseño óptimo, eficiencia, seguridad y dinamismo en las aplicaciones web.

1.5.1. PHP 5.3.10

PHP es un lenguaje orientado al desarrollo de aplicaciones web dinámicas con acceso a información contenida en bases de datos. Es sencillo y fácil de aprender, además de poseer una amplia documentación en su sitio web oficial. Entre sus características esenciales destaca el hecho de ser libre, multiplataforma, permitir el manejo de excepciones y aplicar técnicas de la programación orientada a objetos.

PHP 5 es la versión más reciente de un lenguaje útil para la creación de aplicaciones web; se trata de una renovación total que se ha extendido a numerosos aspectos, como la interacción con otras tecnologías y la sintaxis de orientación a objetos. Independientemente de sus potencialidades es imprescindible en caso

de la utilización del *framework* de desarrollo Symfony.

1.5.2. HTML 5

HTML es el lenguaje de marcado predominante en el desarrollo de páginas web; se utiliza en la traducción y descripción de la estructura y la información en forma de texto, así como para complementar este con otros elementos.

HTML 5 es una actualización de HTML, el lenguaje en el que es creada la web. También es un término de marketing para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML 5, CSS 3 y Javascript. Aunque aún es un lenguaje en desarrollo, los principales navegadores, en sus últimas versiones, reconocen muchos de los elementos que aporta. Entre sus características resalta la reducción de la necesidad de *plugins* externos y un mejor manejo de errores.

1.5.3. CSS 3

CSS es un lenguaje para definir el estilo o la apariencia de las páginas web escritas con HTML o de los documentos XML⁴². Concede separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas web. Su novedad más importante consiste en la incorporación de mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos que a menudo complicaban el código.

1.5.4. JavaScript

JavaScript es un lenguaje de programación interpretado que se utiliza principalmente para crear páginas web dinámicas. Permite mejoras en la interfaz de usuario y páginas web para agregar funcionalidad, validar formularios y comunicarse con el servidor. Es un lenguaje basado en prototipos, imperativo y dinámico. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript.

1.5.5. Justificación de la selección

Las aplicaciones web son una de las tendencias modernas en el desarrollo de software. El uso de lenguajes para este tipo de implementaciones resulta indispensable para el éxito del presente trabajo. La selección de los lenguajes anteriores se sustenta en que:

- ✓ El uso de Symfony como *framework* de desarrollo deriva que sean empleados los lenguajes de

⁴² Siglas del Inglés *Extensible Markup Language* (Lenguaje de Marcas Extensible), es un metalenguaje extensible de etiquetas, una manera de definir lenguajes para diferentes necesidades.

programación: PHP, HTML, CSS y JavaScript.

- ✓ PHP 5.3.10: Symfony es un *framework* para PHP y este lenguaje proporciona a la propuesta de solución un alto grado de dinamismo y seguridad.
- ✓ HTML 5 permite definir la estructura y el contenido de las páginas, permitiendo combinar textos, imágenes, sonidos, vídeos y enlaces a otras páginas.
- ✓ CSS 3 permite la separación entre las normas de presentación y el contenido a mostrar en la aplicación.
- ✓ JavaScript es el encargado de interpretar numerosas instrucciones y ejecutarlas para realizar efectos y permitir la interacción con las páginas web de una manera más dinámica.

1.6. Servidor *streaming*⁴³

La tecnología de *streaming* se utiliza para aligerar la descarga y ejecución de audio y vídeo en la Web, ya que permite escuchar y visualizar los archivos mientras se están descargando. Aunque en muchas ocasiones no es necesario utilizar un servidor *streaming* pueden ofrecer importantes prestaciones como mandar un archivo de mayor o menor calidad dependiendo de la velocidad de la línea. Para la transmisión de un evento en directo, se hace imprescindible contar con un servidor de *streaming* al que se mandará la señal y con ella, la enviará a todos los clientes a medida que la va recibiendo.

1.6.1. Flumotion 0.10.0

Flumotion es un servidor de medios de *streaming* que se distribuye bajo la licencia LGPL. Cuenta con herramientas gráficas de administración intuitiva, haciendo que la tarea de crear y manipular flujos de audio y de vídeo sea fácil para los administradores de sistemas, incluso principiantes. (30)

Flumotion soporta actualmente el *streaming* de cámaras web, tarjetas de captura de TV, FireWire⁴⁴ y cámaras DV⁴⁵. Cuenta con un diseño distribuido, por lo que es fácil de repartir la carga en varios equipos, lo que le permite hacer la manipulación mediática avanzada. Tiene la ventaja de contar con una empresa comercial que lo respalda lo que significa que posee soporte comercial y funcionalidades adicionales disponibles. Cuenta con soporte para una amplia gama de dispositivos de entrada de audio y vídeo

⁴³ Es la distribución de multimedia a través de una red de computadoras de manera que el usuario consume el producto al mismo tiempo que se descarga.

⁴⁴ Estándar de interfaz de conexión para diversas plataformas, destinado a la entrada y salida de datos en serie a gran velocidad.

⁴⁵ Siglas del Inglés *Digital Video* (Vídeo Digital) es un estándar de vídeo de gama doméstica, industrial y broadcast

compatibles con el sistema operativo GNU/Linux⁴⁶. La selección se basa en el hecho de ser libre, y que su servicio de *streaming* se realiza a través del protocolo HTTP⁴⁷.

1.7. Entornos Integrados de Desarrollo (IDE⁴⁸)

Un IDE es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios. Los IDE proveen un entorno de desarrollo amigable para el uso de la mayoría de los lenguajes de programación.

1.7.1. NetBeans 7.3

NetBeans es un entorno de desarrollo integrado disponible para Windows⁴⁹, Mac⁵⁰, GNU/Linux y Solaris⁵¹. Consiste en un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente aplicaciones web, de escritorio y móviles utilizando la plataforma Java, así como PHP, JavaScript y otros. Puede ser integrado fácilmente con Symfony 2; soporta PHP 5, CCS 3 y HTML5. Está apoyado por una comunidad de desarrolladores y ofrece una amplia documentación, así como una gran cantidad recursos de capacitación.

El IDE NetBeans 7.3 ofrece un rendimiento significativamente mejorado y mayor experiencia de codificación. La base en la que se sustenta su elección es que permite desarrollar aplicaciones utilizando el *framework* Symfony y ejecutar los comandos del mismo directamente desde la interfaz del IDE.

1.8. Sistemas Gestores de Bases de Datos (SGBD⁵²)

Un SGBD consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a ellos. El objetivo primordial de un SGBD es proporcionar un entorno que sea conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos.

1.8.1. PostgreSQL 9.1

PostgreSQL es un SGBD objeto relacional, distribuido bajo licencia BSD⁵³ y con código fuente disponible

⁴⁶ Términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux con el sistema GNU.

⁴⁷ Siglas del Inglés *Hypertext Transfer Protocol* (Protocolo de Transferencia de Hipertexto), protocolo usado en cada transacción de la World Wide Web.

⁴⁸ Siglas del Inglés de *Integrated Development Environment*

⁴⁹ Microsoft Windows es el nombre de una familia de sistemas operativos desarrollados y vendidos por Microsoft.

⁵⁰ Sistema operativo creado por Apple para su línea de computadoras Macintosh.

⁵¹ Sistema operativo de tipo Unix.

⁵² Siglas del Inglés *Data Base Management System*.

⁵³ Licencia de software otorgada principalmente para los sistemas *Berkeley Software Distribution* (BSD).

libremente. PostgreSQL utiliza un modelo cliente/servidor y multiprocesos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (31)

Entre las características importantes de este SGBD se tiene la eliminación de obstáculos para el despliegue de nuevas aplicaciones o portado en PostgreSQL. Además incluye replicación sincrónica que permite alta disponibilidad con consistencia a través de varios servidores: admite ordenación lingüística por base de datos, tabla o columna y mejora en gran medida el rendimiento de datos efímeros. Tiene la capacidad de conectarse y consultar otras bases de datos de PostgreSQL, así como de crear fácilmente, cargar y administrar las características de nuevas bases de datos. Además de ser un SGBD de código abierto, las principales causas por las cuales se propone su utilización consisten en que:

- ✓ El núcleo de la aplicación web para el almacenamiento de audio y vídeos digitales ya hace uso de este SGBD.
- ✓ Aparece documentado en la base tecnológica del centro de desarrollo al cual responde la solución.
- ✓ Cuenta con una amplia documentación y una comunidad de desarrollo dentro de la UCI.
- ✓ Junto a MySQL es uno de los SGBD más difundidos, razón por la cual el usuario final solicitó su utilización.

1.9. Ingeniería del Software asistida por computadoras (CASE⁵⁴)

Las herramientas CASE, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste del mismo en términos de tiempo y dinero. Estas herramientas pueden ser útiles en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como la realización de un diseño de proyecto, cálculos de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.

1.9.1. Visual Paradigm

Visual Paradigm es una herramienta CASE que proporciona un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Es capaz de soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de diferentes tipos de diagramas.

Esta herramienta está disponible en múltiples plataformas (Windows, GNU/Linux), su diseño se centra en casos de usos y se enfoca a la calidad del software. Es fácil de instalar y actualizar, es distribuida bajo

⁵⁴ Siglas del Inglés *Computer Aided Software Engineering*.

licencia libre, tiene capacidad de ingeniería directa e inversa, soporta UML⁵⁵ y ORM, y sus versiones son compatibles entre sí. Brinda funcionalidades para la generación de código fuente a partir de diagramas de clases. Está diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos.

1.10. Metodología de desarrollo

El empleo de una metodología durante el desarrollo de software, le confiere a este y a la investigación que se le asocia, transparencia y calidad, elementos muy perseguidos por el cliente y necesarios para el producto. La aplicación correcta de una metodología ayuda a optimizar el tiempo de desarrollo del software, a definir con exactitud el personal que requiere el proyecto, las herramientas a utilizar, conocimientos concretos sobre el problema a resolver, entre otros aspectos que deben ser chequeados y controlados.

1.10.1. OpenUP

OpenUP es un proceso ágil y unificado, que contiene el conjunto mínimo de prácticas a seguir por un equipo para el desarrollo eficaz del software. Es un proceso iterativo e incremental que es mínimo, completo y extensible, fácilmente adaptable a una amplia variedad de proyectos. Se distingue por estar centrado en la arquitectura y guiado por casos de uso.

OpenUP se caracteriza, además por cuatro principios básicos que se soportan mutuamente:

1. Colaboración para alinear los intereses y un entendimiento compartido.
2. Balance para confrontar las prioridades (necesidades y costos técnicos) y maximizar el valor para los *stakeholders*⁵⁶.
3. Enfoque hacia la articulación de la arquitectura para facilitar la colaboración técnica, reducir los riesgos y minimizar excesos y trabajo extra.
4. Evolución continua para reducir riesgos, demostrar y obtener resultados. (32)

Se propone el uso de OpenUP debido a que:

- ✓ Es una metodología de código abierto, diseñada para pequeños equipos de desarrollo en los cuales se valora la colaboración y el aporte de los *stakeholders* sobre los entregables y las

⁵⁵ Siglas del inglés *Unified Modeling Language* (Lenguaje Unificado de Modelado). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

⁵⁶ En español, interesados. Se refiere a personas y organizaciones como clientes, patrocinadores, organización ejecutante y el público, involucrados activamente con el proyecto, o cuyos intereses pueden verse afectados de manera positiva o negativa por la ejecución o conclusión del proyecto.

formalidades innecesarias.

- ✓ No define necesariamente modelos de negocio ni dominio y permite detectar errores tempranos a través de un ciclo iterativo.
- ✓ Promueve un enfoque centrado en el cliente y un ambiente colaborativo entre los miembros del equipo de desarrollo.
- ✓ El centro de desarrollo al cual responde la presente investigación ha solicitado el empleo de la misma como parte de un programa de mejora.

1.11. Conclusiones parciales

Durante la evaluación de las tendencias de aplicaciones similares fueron identificados sistemas con un alto nivel de aceptación, que aunque no satisfacen ni las expectativas del proyecto para realizar un uso correcto de la gestión documental, ni las necesidades y restricciones del cliente en cuanto a la representación y manejo de audio y vídeos digitales, permitieron adquirir nuevos elementos que tributan al mejoramiento del sistema.

En la selección de las herramientas y tecnologías para el desarrollo de la solución fue tomada en cuenta la política de uso de herramientas con soporte multiplataforma y basadas en software libre, las cuales fueron escogidas por el cliente y el proyecto. Por tanto se utilizó como *framework* de desarrollo Symfony 2.1.4, jQuery 1.9.2 y Bootstrap 2.2.2 ; como lenguajes para la Web PHP 5, HTML 5 , CSS 3 y JavaScript; como SGBD PostgreSQL 9.1; como herramienta CASE para el modelado del Software Visual Paradigm 8.0 y como metodología de desarrollo OpenUP. Para apoyar el proceso de implementación fue escogido el IDE de desarrollo NetBeans 7.3.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DE LA APLICACIÓN

La parte más difícil en la construcción de sistemas es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan ardua como establecer los requerimientos técnicos detallados. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si se realiza de forma errónea. Ninguna otra parte es tan difícil de rectificar posteriormente. (33)

En este sentido se propone el presente capítulo de la investigación, tomando el modelo de análisis como la primera representación técnica de la aplicación para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0. Se redireccionan esfuerzos hacia la ejecución de una serie de tareas de modelado que llevan a una especificación completa de los requisitos y a una representación del diseño general del software a construir.

2.1. Propuesta de solución

La aplicación web TeVeo Plus V1.0, es un conjunto de soluciones informáticas para la Web, basada en estándares abiertos, para la gestión de los contenidos en el Sistema de Gestión Documental de Audio y Vídeos Digitales, cuya finalidad es permitir o facilitar la identificación, descripción, catalogación, recuperación y publicación de los recursos multimedia, además permite generar automáticamente diferentes perfiles a partir del archivo de audio o vídeo adicionado originalmente al sistema.

TeVeo Plus V1.0 debe permitir gestionar audio, vídeos, unidades documentales, galerías temáticas y carpetas. La arquitectura de información del sistema deberá ser amigable para el usuario brindando efectividad, eficiencia y satisfacción durante el proceso de gestión documental de multimedia en cualquier institución. A continuación se observa el ciclo de vida de un archivo de audio o vídeo en el sistema (Ver Figura 1), donde intervienen los grupos de usuarios colaborador, editor, gestor e invitado.

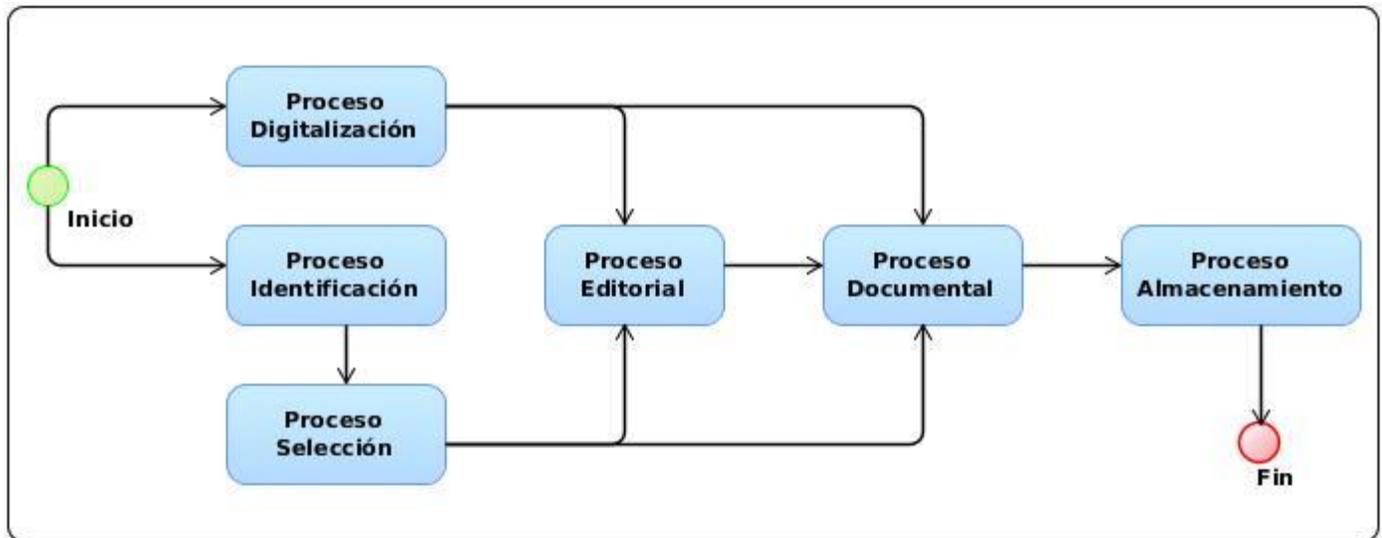


Figura 1. Ciclo de vida de un audio o vídeo en el sistema.

A continuación se describe cada uno de los procesos ilustrados en la figura anterior.

Proceso Identificación: Son identificados los archivos de audio y vídeos digitales. Estos archivos son cargados al sistema por los gestores o por los colaboradores a través del protocolo HTTPS⁵⁷, garantizando que sean archivados para ser procesados solo aquellos archivos multimedia con extensión webm⁵⁸, ogg⁵⁹, ogx⁶⁰ u ogv⁶¹. Una vez recepcionados dichos materiales, el colaborador o el gestor pueden llenar un grupo de campos que se corresponden con las características identificativas del archivo y del acontecimiento que ilustran; posteriormente se generará de modo automático una miniatura asociada al material mediante un identificador (id), que será almacenada durante el proceso de almacenamiento. Finalmente el proceso ofrecerá la posibilidad de descargar, visualizar o enviar al Proceso de Selección la multimedia añadida (Ver Anexo 1).

Proceso Digitalización: Son digitalizados los archivos multimedia. Es el proceso inicial del ciclo de vida de una multimedia. Tiene como objetivo obtener de una fuente fiable, el gestor, la información relativa al archivo multimedia, mediante los atributos identificativos, que constituyen los elementos que servirán para la descripción posterior de los archivos multimedia. El gestor envía la multimedia al proceso editorial o documental.

⁵⁷ Siglas del Inglés *Hypertext Transfer Protocol Secure* (Protocolo Seguro de Transferencia de Hipertexto), protocolo destinado a la transferencia segura de datos de Hipertexto.

⁵⁸ Formato multimedia abierto y libre desarrollado por Google y orientado a usarse con HTML5.

⁵⁹ Formato contenedor, es además el contenedor nativo para los códecs multimedia que desarrolla Xiph.Org.

⁶⁰ Extensión de archivo de audio.

⁶¹ Extensión de archivo multimedia.

Proceso Selección: Este proceso es el encargado de decidir el tratamiento que se le dará a los archivos multimedia dentro de la aplicación. Una vez recepcionada la multimedia, si esta no ha sido editada se le adicionan los aspectos de selección; en caso de que ya haya sido editada simplemente se procede a descargar, visualizar o enviar el audio o vídeo a los siguientes procesos y espacios:

- ✓ Proceso Editorial si es necesario que el material sea editado o publicado con inmediatez.
- ✓ Proceso Documental si es necesario documentarlo.
- ✓ Papelera si ha sufrido daños y no cuenta con la calidad requerida (Ver Anexo 1).

Proceso Editorial: Este proceso incluye dos subprocesos: Edición (encargado de las modificaciones que necesiten el audio o vídeo) y Publicación (encargado de publicar el material en caso de ser necesario e inmediato). Dicho de otro modo, este proceso permite, haciendo uso del subproceso Edición, una vez recepcionada, editar la multimedia y enviarla al Proceso Selección donde será revisada y aprobada o no para ser publicada por el editor general; o decidir su publicación haciendo uso del subproceso Publicación, permitiendo visualizarla, descargarla o enviarla al Proceso Documental para ser documentada (Ver Anexo 1).

Proceso Documental: Este proceso, cuando la multimedia es recepcionada, se encarga de documentarla, permitiendo que el gestor le añada atributos conceptuales. Posteriormente ofrece la posibilidad de descargar, visualizar o enviar la multimedia al Proceso Almacenamiento (Ver Anexo 1).

Proceso Almacenamiento: Es la forma en que se disponen los archivos de audio y vídeos. El objetivo es organizarlos físicamente para su recuperación efectiva, una vez que hayan sido localizados. Estos son organizados atendiendo a las colecciones y sus objetivos (Ver Anexo 1).

El sistema que se propone en esta solución presenta mejoras significativas de usabilidad entre las que cabe destacar las siguientes:

Visibilidad del sistema: Cuando se selecciona un ícono se diferencia de los no seleccionados. La terminología del menú es constante en todo el sitio. Los enlaces del menú se resaltan cuando se seleccionan.

Adecuación del sistema: Los íconos que aparecen se identifican claramente con lo que representan. No es necesaria la experiencia del usuario para navegar por la web. La información que se presenta en la aplicación es fácil de entender y memorizar. La información está organizada con categorías lógicas, fácilmente memorizables para el usuario. El usuario puede ponerse en contacto con el encargado del Sitio web (correo electrónico del administrador) para hacer sugerencias o comentarios. Los mensajes de error están en texto plano, entendible. La información es de rápida lectura, y con una disposición asequible. El

formato de fechas empleado es correcto. Existe una manera rápida y sencilla de acceder a páginas relacionadas o a otras secciones. El nombre de los botones de un formulario es adecuado, aplicado a la acción.

Control y libertad de usuarios: Si la respuesta se retrasa, aparece un mensaje como que el sistema está procesando la acción. La página se ve con cualquier resolución de pantalla.

Consistencia y estándares: La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación. Un mismo elemento aparece igual en todo el sitio. Se usan nombres estandarizados (“mapa web”, “acerca de”). Una imagen que sirve como enlace es fácilmente distinguible.

Prevención de errores: Se dan indicaciones para completar campos problemáticos. En situaciones donde se pueden producir errores de escritura existe la posibilidad de seleccionar la información de una lista.

Reconocimiento: La estructura resulta evidente para el usuario. La manera de navegar por la web o ejecución de tareas asignadas se aprenden de forma rápida. No es necesaria la experiencia del usuario para navegar por la web. Se identifican fácilmente las figuras, las tablas, los enlaces, las zonas activas y el tipo de acción que se debe ejecutar. Los íconos asociados con contenido se relacionan fácilmente.

Flexibilidad y eficiencia de uso: Las tareas repetitivas se efectúan con facilidad. Se evitan los pasos inútiles. Las partes o secciones más importantes de los sitios son accesibles desde la página de inicio.

Estética: El menú de navegación aparece en un lugar destacado. La información importante aparece destacada.

2.2. Especificación de Requisitos de Software

Los requisitos surgen como ideas que pueden originarse como una respuesta a una amenaza percibida o competencia del mercado, de una imposición legal, del deseo de crear un nuevo sistema, de la necesidad de reemplazar un sistema existente o alguna otra necesidad percibida. La Especificación de Requisitos de Software es una descripción completa del comportamiento del sistema que se va a desarrollar.

La meta principal en esta etapa es identificar y documentar lo que en realidad se necesita, de una forma en que pueda fácilmente ser transmitido al cliente y al equipo de desarrollo. Para el desarrollo de TeVeo Plus V1.0 se realizaron reuniones con un representante del cliente en las que se obtuvieron las características funcionales y no funcionales que el cliente desea que cumpla la aplicación.

2.2.1. Requisitos funcionales

Los requisitos funcionales definen el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. Los requisitos funcionales de la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0 son descritos a continuación.

Requisitos funcionales

Usuarios

- RF_1. Autenticar usuario
- RF_2. Adicionar usuario
- RF_3. Editar usuario
- RF_4. Eliminar usuario
- RF_5. Deshabilitar cuenta usuario
- RF_6. Visualizar cuentas de usuarios

Perfil del usuario

- RF_7. Visualizar perfil del usuario
- RF_8. Visualizar cuota del usuario
- RF_9. Cambiar contraseña del usuario

Grupos

- RF_10. Buscar grupos
- RF_11. Adicionar grupos
- RF_12. Eliminar grupos
- RF_13. Asignar usuario a grupo(s)
- RF_14. Eliminar usuarios del grupo(s)

Contenidos

- RF_15. Configurar contenidos
- RF_16. Adicionar contenidos
- RF_17. Editar contenidos

- RF_18. Eliminar contenidos
- RF_19. Descargar contenidos
- RF_20. Arrastrar y pegar contenidos
- RF_21. Visualizar contenidos
- RF_22. Votar contenido

Comentarios

- RF_23. Adicionar comentarios
- RF_24. Visualizar comentarios
- RF_25. Eliminar comentarios

Búsquedas

- RF_26. Realizar búsqueda simple
- RF_27. Realizar búsqueda avanzada

Comunes

- RF_28. Visualizar ayuda online
 - RF_29. Visualizar contáctenos
 - RF_30. Visualizar ruta de navegación
-

2.2.2. Requisitos no funcionales

Un requisito no funcional es un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema, en lugar de sus comportamientos específicos. Por tanto, se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar. Los requisitos no funcionales de la aplicación web para el Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0 son descritos a continuación.

Requisitos No Funcionales

Usabilidad

RNF_US1. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de la Web.

Seguridad

RNF_SEG1. Controlar los permisos de acceso, escritura, lectura en dependencia del rol que desempeñe cada usuario del sistema.

RNF_SEG2. El sistema permitirá que un usuario solamente esté autenticado una vez (sesión única).

RNF_SEG3. El sistema cerrará la sesión de un usuario inactivo por más de 15 minutos.

Rendimiento

RNF_REN1. El sistema requiere un rendimiento eficiente que se apoye en una transferencia mínima de datos cliente-servidor.

Apariencia o Interfaz Externa

RNF_APA_IE1. El sistema debe poseer una interfaz amigable al usuario, brindando facilidades que permitan interactuar y navegar con el sistema de forma fácil y rápida, y con un conocimiento básico de computación.

Software

RNF_SW1. Usar preferentemente para los servidores el sistema operativo Ubuntu 12.04.2 LTS.

RNF_SW2. El sistema debe desarrollarse con la tecnología Symfony 2.1.4, utilizando la arquitectura Modelo Vista Controlador.

RNF_SW3. El lenguaje de programación utilizado será PHP 5.

RNF_SW4. Como gestor de base de datos se utilizará PostgreSQL V9.1.

RNF_SW5. El servidor de aplicaciones será Apache 2.

Restricciones del diseño y la implementación

RNF_RES_DIS1. Las interfaces destinadas al usuario, deben de programarse en PHP y ser compatible con las versiones de los navegadores Microsoft Internet Explorer 10, Mozilla Firefox 19, y Chromium 20.

RNF_RES_DIS2. Las computadoras clientes deben tener un navegador web (recomendado Mozilla Firefox 19).

RNF_RES_DIS3. Implementar el producto con herramientas *Open Source*, sobre la filosofía de Software Libre.

RNF_RES_DIS4. Solo se procesarán audio y vídeos con los siguientes formatos: webm, ogg, ogx y ogv.

Hardware

RNF_HW1. Servidores con al menos: 2 GB RAM⁶², Core i3 3.1, HDD (250-500 GB), soporte para RAID MODE⁶³. (Recomendado una SAN⁶⁴)

Ayuda y documentación en línea.

RNF_DOC_AY1. Debe documentarse la forma de utilizar el sistema a través de un manual de usuario del producto final.

RNF_DOC_AY2. La documentación debe ser indispensable como evidencia de la evolución del Proyecto.

2.2.3. Diagrama de Casos de Uso del Sistema

El diagrama de casos de uso del sistema representa la interacción del actor o los actores con los procesos del sistema. (Ver Figura 2)

⁶² Siglas del Inglés *Random-Access Memory*, (Memoria de Acceso Aleatorio) se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del software.

⁶³ Modo que proporciona la mayor flexibilidad y capacidad para actualización general debido a que permite que el sistema esté listo para soportar RAID, sistema de almacenamiento que usa múltiples discos duros entre los que se distribuyen o replican los datos.

⁶⁴ Siglas del Inglés *Storage Area Network* (Red de Área de Almacenamiento), es una red de almacenamiento integral.

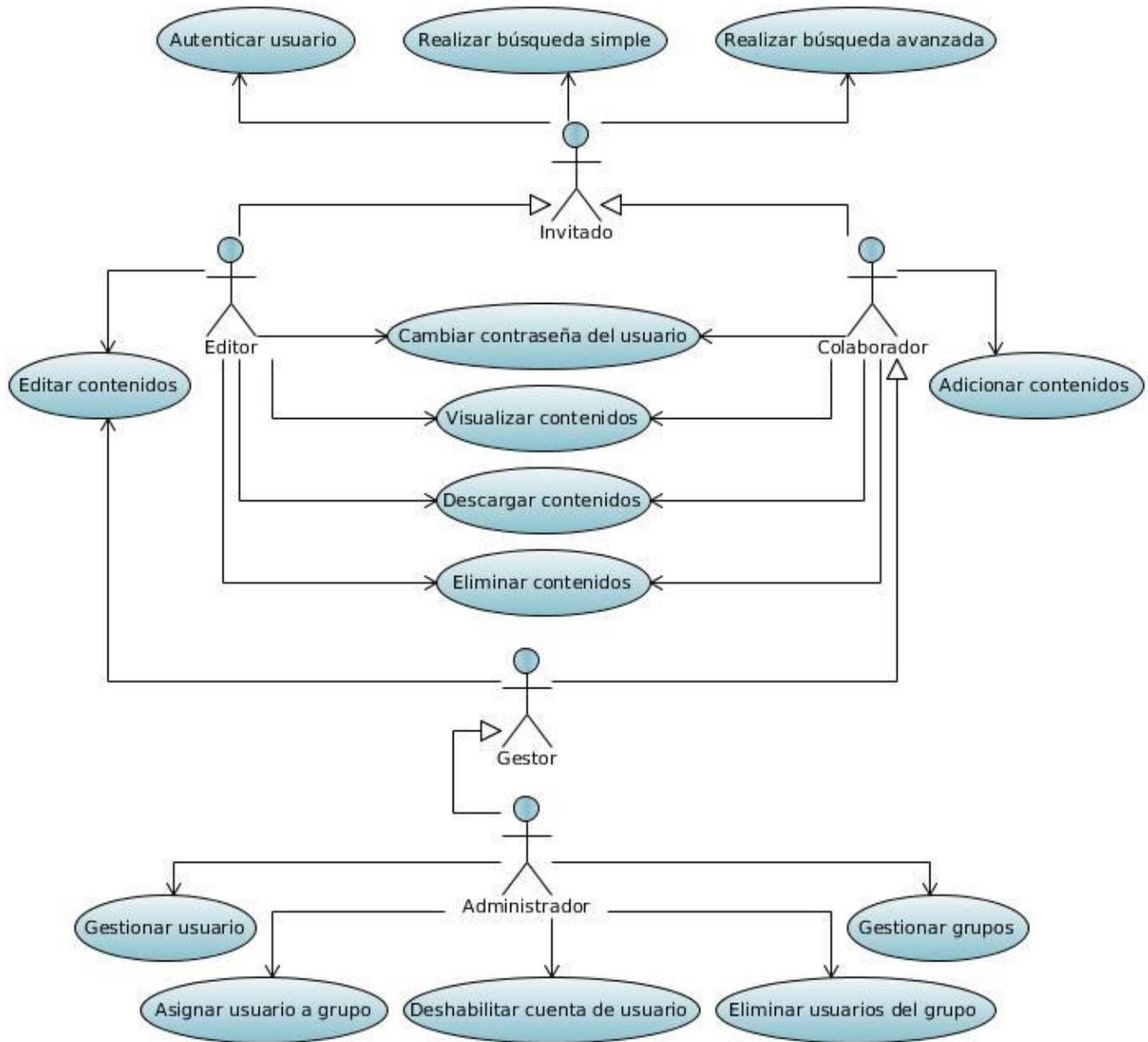


Figura 2. Diagrama de Casos de Uso del sistema.

2.2.4. Especificación de Casos de Uso

Descripción de los actores

Los actores de un Caso de Uso representan a usuarios u otros sistemas que interactúan con el sistema. El nombre que se le dé a un actor debe reflejar el papel que tiene este en el sistema. Mayormente los actores representan un tipo de usuario, dígame rol, que puede ser ocupado por varias personas a la vez. Esto significa que cuando una persona interactúa con el sistema de diferentes maneras (asumiendo diferentes roles), estará representado por varios actores.

Invitado	Persona, de cualquier índole, interesada en acceder al sistema que puede visualizar información pero sin acceso a los procesos antes de autenticarse.
Colaborador	Responsable de proveer de información a la aplicación.
Editor	Responsable de editar el archivo multimedia, dígase descargarlo, realizarle las modificaciones pertinentes y subirlo al sistema nuevamente.
Gestor	Responsable de proveer información al sistema, pero no posee autoridad para supervisar o tomar decisiones importantes.
Administrador	Responsable de supervisar y tomar decisiones necesarias y pertinentes en cuanto a la gestión de todo tipo de contenido que se maneja en la aplicación, así como permisos y gestión de recursos.

Descripción de Casos de Uso

CU_1. Autenticar Usuario

Objetivo	Autenticación en el sistema.	
Actores	Administrador: (Inicia) se autentica con su usuario y contraseña en el sistema. Colaborador:(Inicia) se autentica con su usuario y contraseña en el sistema. Editor:(Inicia) se autentica con su usuario y contraseña en el sistema. Gestor:(Inicia) se autentica con su usuario y contraseña en el sistema.	
Resumen	El CU permite el acceso al sistema mediante la inserción de usuario y contraseña según el rol.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Los datos introducidos han sido validados.	
Postcondiciones	Usuario autenticado en su espacio personal.	
Flujo de eventos		
Flujo básico Autenticar usuario.		
	Actor	Sistema
1.	Necesita autenticarse en el sistema. Para hacerlo, abre su navegador y escribe la dirección URL del Sistema de Gestión Documental de Audio y Vídeos Digitales TeVeo Plus V1.0 (http://teveoplus.uci.cu).	Muestra la página de inicio de la aplicación en la que aparece en la parte superior derecha un formulario de autenticación con los siguientes campos: Usuario* (TEXT) (20 caracteres): Contraseña* (PASSWORD) (20 caracteres):
2.	Introduce datos de autenticación y presiona el	Valida que los datos están completos.

	botón que se encuentra a la derecha del formulario para enviar los datos al sistema.	Valida que los datos sean correctos.
3.		Permite el acceso al sistema mostrando el espacio personal en dependencia del usuario autenticado.
4.		Termina el caso de uso.
Flujos alternos		
1. Los datos son incorrectos		
	Actor	Sistema
1.	Ha introducido los campos usuario o contraseña de forma incorrecta.	Mostrará un mensaje de error: Si el usuario introducido es incorrecto mostrará un mensaje de error que indicará que el usuario o la contraseña son incorrectos y continúa en la página inicial. Si la contraseña introducida es incorrecta mostrará un mensaje de error que indicará que el usuario o la contraseña son incorrectos y continúa en la página inicial.
2.		Termina el caso de uso.
2. Los campos están vacíos.		
	Actor	Sistema
1.	Ha dejado campos vacíos a la hora de autenticarse.	Mostrará un mensaje de error: Si el campo "Nombre de Usuario" está vacío mostrará un mensaje de error que le señala al usuario que ese campo no puede estar vacío y continúa en la página inicial. Si el campo "Contraseña" está vacío mostrará un mensaje de error que le señala al usuario que ese campo no puede estar vacío y continúa en la página inicial.
2.		Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos funcionales	no	
Asuntos pendientes		

Tabla 1. Especificación de requisitos CU Autenticar Usuario.

Las descripciones de los casos de usos Adicionar contenido, Editar contenido y Eliminar contenido pueden ser consultadas en el Anexo 2.

2.3. Descripción de estilos arquitectónicos y patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Además identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades. Entre las definiciones más comunes se encuentra la de ser una solución estándar para un problema común de programación, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios o un proyecto o estructura de implementación que logra una finalidad determinada.

2.3.1. Patrones arquitectónicos

Especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. La propuesta de solución para el sistema hará uso del patrón Modelo-Vista-Controlador.

Modelo-Vista-Controlador (MVC)

Symfony basa su funcionamiento interno en la arquitectura Modelo - Vista – Controlador (MVC), utilizada por la mayoría de *frameworks* web. La utilización de dicho marco de trabajo para el desarrollo del producto conlleva a la implementación de este patrón. El siguiente esquema simplificado muestra el funcionamiento interno de Symfony (Ver Figura 3).

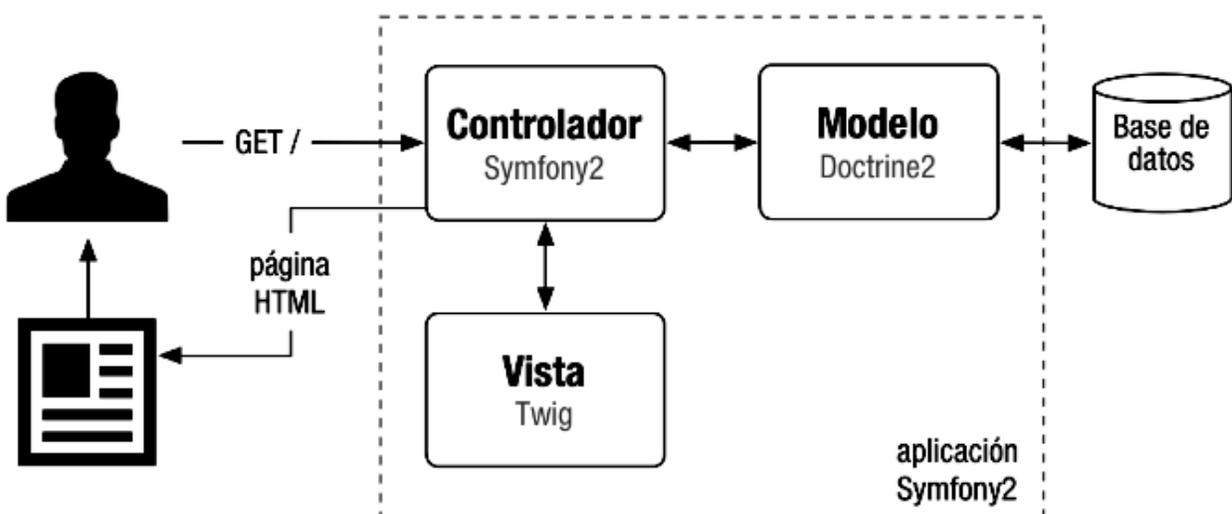


Figura 3. Esquema simplificado de la arquitectura interna de Symfony. (34)

Cuando el usuario solicita ver la portada del sitio, internamente sucede lo siguiente:

1. El sistema de enrutamiento determina que Controlador está asociado con la página de la portada.
2. Symfony ejecuta el Controlador asociado a la portada.
3. El Controlador solicita al Modelo los datos de la petición.
4. Con los datos devueltos por el Modelo, el Controlador solicita a la Vista que cree una página mediante una plantilla y que inserte los datos del Modelo.
5. El Controlador entrega al servidor la página creada por la Vista.

Con Symfony se pueden llegar a hacer cosas muy complejas, pero el funcionamiento interno siempre es el mismo:

- 1) El Controlador manda y ordena.
- 2) El Modelo busca la información que se le pide.
- 3) La Vista crea páginas con plantillas y datos.

La arquitectura MVC proporciona ventajas, como la organización del código, la reutilización y la flexibilidad. Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo.

2.3.2. Patrones de diseño

Especifican un conjunto de aspectos relacionados con el diseño de los subsistemas, por tanto se centran en aspectos más específicos. Últimamente hay múltiples esfuerzos enfocados a la construcción de herramientas de desarrollo basados en los patrones y frecuentemente se incluye en los nombres de las clases el nombre del patrón en que se basan, facilitando así la comunicación entre desarrolladores.

Patrones GRASP⁶⁵

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. (35)

Experto

Permite asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

⁶⁵ Siglas del Inglés *General Responsibility Assignment Software Patterns* (Patrones Generales de Software para Asignar Responsabilidades)

Establece el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto o la implementación de un método debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

En Symfony 2 este patrón es utilizado en la inclusión de Doctrine para mapear la base de datos. Se utiliza para crear una capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases en todas las funcionalidades comunes a las entidades; las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla asociada.

Creador

Su implementación permite identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que:

- ✓ Tiene la información necesaria para realizar la creación del objeto.
- ✓ Usa directamente las instancias creadas del objeto.
- ✓ Almacena o maneja varias instancias de la clase.
- ✓ Contiene o agrega la clase.

Puede soportar un bajo acoplamiento, mayor claridad y reutilización. Este patrón es utilizado en la implementación de las clases controladoras donde se encuentran las acciones definidas para el sistema, ejecutándose en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades.

Alta cohesión

Resuelve el problema de cómo mantener la complejidad dentro de límites manejables. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Permite asignar una responsabilidad de modo que la cohesión siga siendo alta.

Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. 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.

Bajo acoplamiento

Resuelve el problema de cómo dar soporte a una dependencia escasa y a un aumento de la reutilización. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Acoplamiento bajo significa que una clase no depende de muchas clases. Permite asignar una responsabilidad para mantener bajo acoplamiento.

Las clases controladoras heredan únicamente de la clase *Controller* para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

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.

Controlador

Resuelve el problema de quién debería encargarse de atender un evento del sistema. Un evento del sistema es un evento de alto nivel generado por un actor externo; es un evento de entrada externa. Se asocia a operaciones del sistema: las que emite en respuesta a los eventos del sistema.

Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. Permite asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema.

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.

Patrones GoF⁶⁶

Los patrones GoF (del inglés *Gang-of-Four*) “Pandilla de los cuatro” (36), son utilizados para ocultar la complejidad del sistema, mostrando solamente lo necesario para el usuario.

Comando

Regula el comportamiento, a nivel de objetos. Permite definir la acción que se quiere ejecutar como un objeto, al cual pueden referirse todos aquellos elementos gráficos que lo deseen. Su uso es apropiado cuando lo fundamental en la relación entre una petición y la acción que la satisface es la flexibilidad.

⁶⁶ Gang of Four, patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos.

Este patrón, al encapsular en un objeto la acción, promueve una separación entre dicha acción y la petición que resuelve, lo cual redundará en una mayor flexibilidad en todo lo relativo a la acción. Esto tiene las siguientes consecuencias:

- ✓ Diferentes objetos pueden ejecutar la misma acción sin necesidad de repetir su declaración e implementación. Basta con que la compartan.
- ✓ Se puede cambiar con facilidad (incluso dinámicamente) la acción que realiza o está asociada a un objeto.
- ✓ Se pueden añadir nuevas acciones sin tocar las clases ya existentes.
- ✓ Se puede especificar, encolar y ejecutar una acción en momentos diferentes.

Decorador

Permite añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades. En Symfony la vista se separa en una plantilla base y varias plantillas que heredan de esta. Normalmente, la plantilla base es global en toda la aplicación y contiene el código HTML que es común a la mayoría de las páginas, lo cual es una implementación del patrón Decorador. Su uso aporta una mayor flexibilidad que la herencia estática, permitiendo, entre otras cosas, añadir una funcionalidad dos o más veces. Además evita concentrar en lo alto de la jerarquía clases guiadas por las responsabilidades, es decir, que pretenden satisfacer todas las posibilidades. De esta forma las nuevas funcionalidades se componen de piezas simples que se crean y se combinan con facilidad.

Registro

Este patrón es útil para los desarrolladores en la Programación Orientada a Objetos. Es un medio sencillo y eficiente de compartir datos y objetos en la aplicación sin la necesidad de preocuparse por conservar numerosos parámetros o hacer uso de variables globales. Se aplica en la clase de configuración que es la encargada de guardar todas las variables globales del sistema.

Los autores de la presente investigación consideran que los patrones de diseño proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifican y describen formas de solucionar problemas que ocurren de forma frecuente en el desarrollo.

El uso de los patrones de diseño mencionados con anterioridad se sustenta en que el *framework* de desarrollo Symfony los implementa por defecto en los proyectos que lo emplean. Además el uso de patrones proporciona:

- ✓ Elementos reusables en el diseño de sistemas software, lo que significa que son aplicables a diferentes problemas de diseño en distintas circunstancias.
- ✓ Efectividad comprobada en la resolución de problemas similares en ocasiones anteriores.
- ✓ Formalizan un vocabulario común entre diseñadores.
- ✓ Estandarización en el diseño, lo que beneficia notablemente a los desarrolladores.

2.4. Diagrama de Clases del Diseño

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones, utilizándose para modelar la vista de diseño estática de un sistema. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables aplicando ingeniería directa e inversa. El diagrama de clases del diseño para el caso de uso Autenticar Usuario del sistema TeVeo Plus V1.0 se ilustra en la Figura 4.

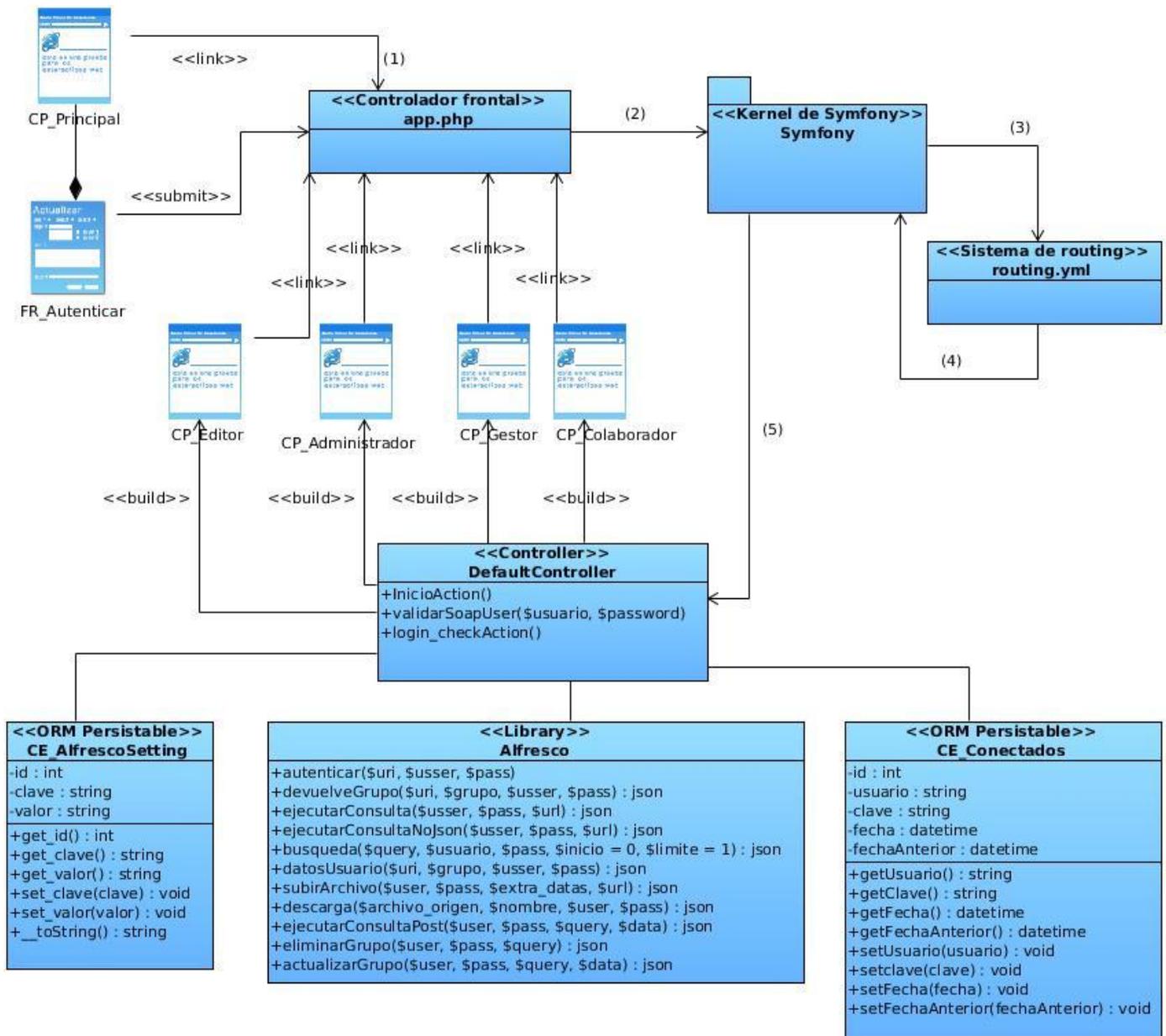


Figura 4. Diagrama de clases del diseño del caso de uso Autenticar Usuario

El uso de Symfony obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por el *framework* según el MVC. El código de la presentación se guarda en la vista, el código de manipulación de datos se guarda en el modelo y la lógica de procesamiento de las peticiones constituye el controlador. La implementación que realiza Symfony de la arquitectura MVC hace que el diseño de clases de cualquier aplicación se adhiera a su estructura. Durante el modelado de clases, para un mejor entendimiento, es necesario reflejar en los diagramas la distribución de los elementos teniendo en cuenta su inclusión dentro de la estructura del *framework*.

Descripción de las clases del diseño

AlfrescoSetting	
Tipo de clase: Entidad	
Atributo	Tipo
id	Int
clave	String
valor	String
Para cada responsabilidad:	
Nombre:	getId()
Descripción:	Funcionalidad encargada de devolver el atributo id.
Nombre:	getClave()
Descripción:	Funcionalidad encargada de devolver el atributo clave, el cual contiene el nombre del valor.
Nombre:	getValor()
Descripción:	Funcionalidad encargada de devolver el atributo valor, el cual contiene las direcciones, espacios y tipos de contenido para la interacción con Alfresco.
Nombre:	setClave()
Descripción:	Funcionalidad que permite modificar el atributo clave.
Nombre:	setValor()
Descripción:	Funcionalidad que permite modificar el atributo valor.
Nombre:	__toString()
Descripción:	Convierte uno o varios atributos de un objeto en una cadena string.

Tabla 2. Clase: AlfrescoSetting.

Alfresco	
Tipo de clase: Librería	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	autenticar(\$uri, \$usser, \$pass)
Descripción:	Funcionalidad encargada de enviar datos de autenticación al Alfresco para que este los valide.
Nombre:	devuelveGrupo(\$uri, \$grupo, \$usser, \$pass)
Descripción:	Funcionalidad encargada de obtener el grupo al que pertenece un usuario.
Nombre:	ejecutarConsulta(\$usser, \$pass, \$url)
Descripción:	Funcionalidad encargada de ejecutar los servicios que proporciona el Alfresco.

Nombre:	ejecutarConsultaNoJson(\$usser, \$pass, \$url)
Descripción:	Funcionalidad que permite ejecutar los servicios que proporciona el Alfresco devolviendo el resultado en un arreglo.
Nombre:	busqueda(\$query, \$usuario, \$pass, \$inicio, \$limite)
Descripción:	Funcionalidad encargada de ejecutar las búsquedas en el Alfresco.
Nombre:	datosUsuario(\$uri, \$grupo, \$usser, \$pass)
Descripción:	Funcionalidad encargada de devolver los datos del usuario.

Tabla 3. Clase: Alfresco.

CE_Conectados	
Tipo de clase: Entidad	
Atributo	Tipo
id	String
usuario	String
clave	String
fecha	Datetime
fechaAnterior	Datetime
Para cada responsabilidad:	
Nombre:	setUsuario(usuario)
Descripción:	Funcionalidad que permite modificar el atributo usuario.
Nombre:	getUsuario()
Descripción:	Funcionalidad encargada de devolver el atributo usuario.
Nombre:	setClave(clave)
Descripción:	Funcionalidad que permite modificar el atributo clave.
Nombre:	getClave()
Descripción:	Funcionalidad encargada de devolver el atributo clave, el cual contiene el identificador único de la sesión.
Nombre:	setFecha(fecha)
Descripción:	Funcionalidad que permite modificar el atributo fecha.
Nombre:	getFecha()
Descripción:	Funcionalidad encargada de devolver el atributo fecha.
Nombre:	setFechaAnterior(fechaAnterior)
Descripción:	Funcionalidad que permite modificar el atributo fechaAnterior.
Nombre:	getFechaAnterior()
Descripción:	Funcionalidad encargada de devolver el atributo fechaAnterior.

Tabla 4. Clase: CE_Conectados.

2.4.1. Diagramas de interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar instancias concretas o prototipos de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. Los diagramas de interacción pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de un conjunto de objetos.

Diagramas de secuencia

Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes. Se conforma colocando en primer lugar los objetos que participan en la interacción en la parte superior del diagrama, a lo largo del eje X. Normalmente, se coloca a la izquierda el objeto que inicia la interacción, y los objetos subordinados a la derecha. A continuación, se colocan los mensajes que estos objetos envían y reciben a lo largo del eje Y, en orden de sucesión en el tiempo, desde arriba hasta abajo. Esto ofrece al lector una señal visual clara del flujo de control a lo largo del tiempo (Ver Figura 5).

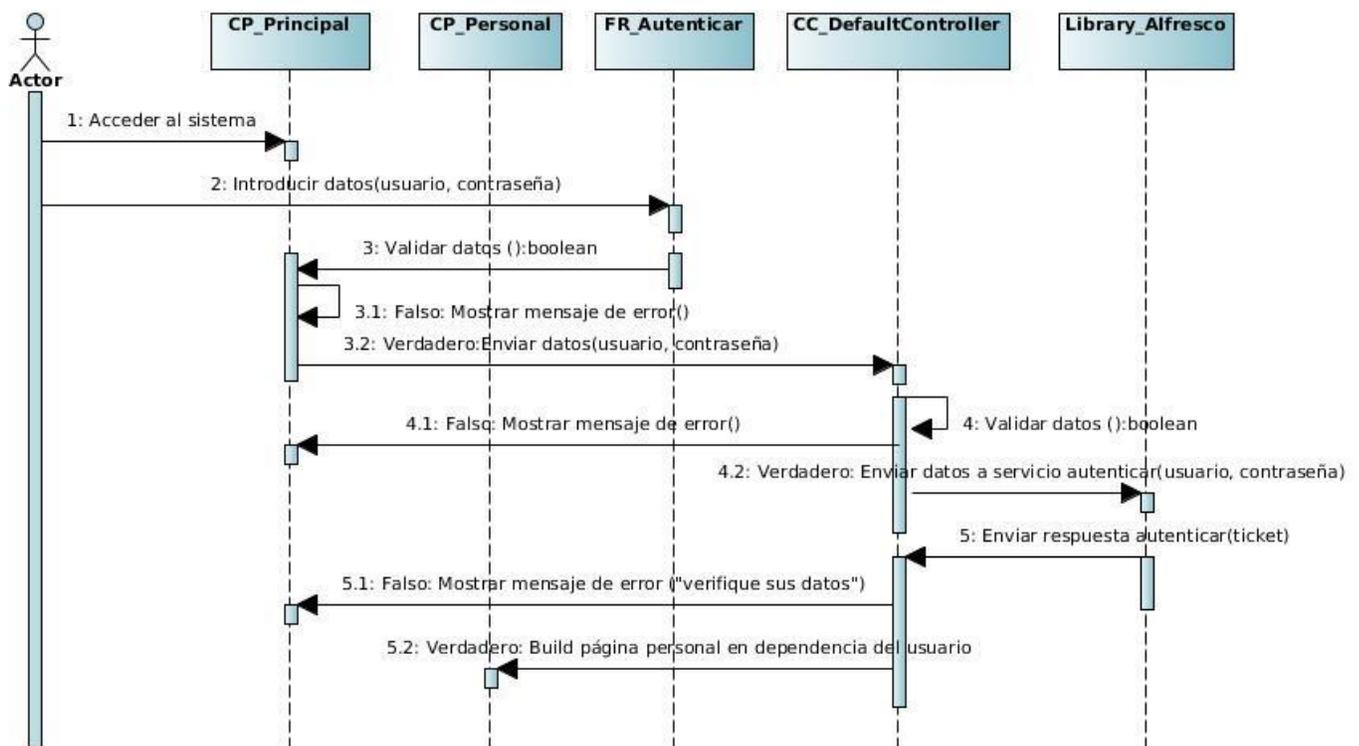


Figura 5. Diagrama de secuencia del caso de uso Autenticar Usuario.

Para el caso de uso autenticar usuario el flujo de eventos del sistema durante el tiempo de ejecución ocurre en la siguiente secuencia:

1. El usuario accede a la Interfaz principal del sistema.
2. El usuario introduce sus datos (usuario y contraseña) en el formulario de autenticación que aparece en la interfaz principal.
3. Los datos introducidos en el formulario de autenticación son validados del lado del cliente en cuanto a longitud y caracteres extraños en el caso del campo usuario.
4. Si los datos han sido validados correctamente del lado del cliente son enviados al controlador, donde nuevamente son validados.
5. Si la validación del lado del servidor es correcta el controlador envía los datos a través del servicio de autenticación que maneja la librería de acceso al Alfresco.
6. La librería envía respuesta al controlador de manera que:
 - 5.1. Si los datos son válidos permite el acceso a la página inicial del espacio que corresponda al rol que solicitó autenticación.
 - 5.2. Si los datos no son correctos envía un mensaje de error en la página inicial del sistema denegando el acceso.

2.5. Diseño de la base de datos

El diseño de la base de datos es la herramienta conceptual para la descripción de datos, relaciones entre ellos, semántica y restricciones de consistencia. Tal representación permite contar con una vista del modelo a implementar a través de la perspectiva de entidades.

2.5.1. Modelo Entidad-Relación de la base de datos

El modelo entidad-relación (E-R) es un modelo de datos de alto nivel. Está basado en una percepción de un mundo real que consiste en una colección de objetos básicos, denominados entidades, y de relaciones entre estos objetos. En la Figura 6 se ilustra el modelo de datos de la aplicación web TeVeo Plus V1.0.

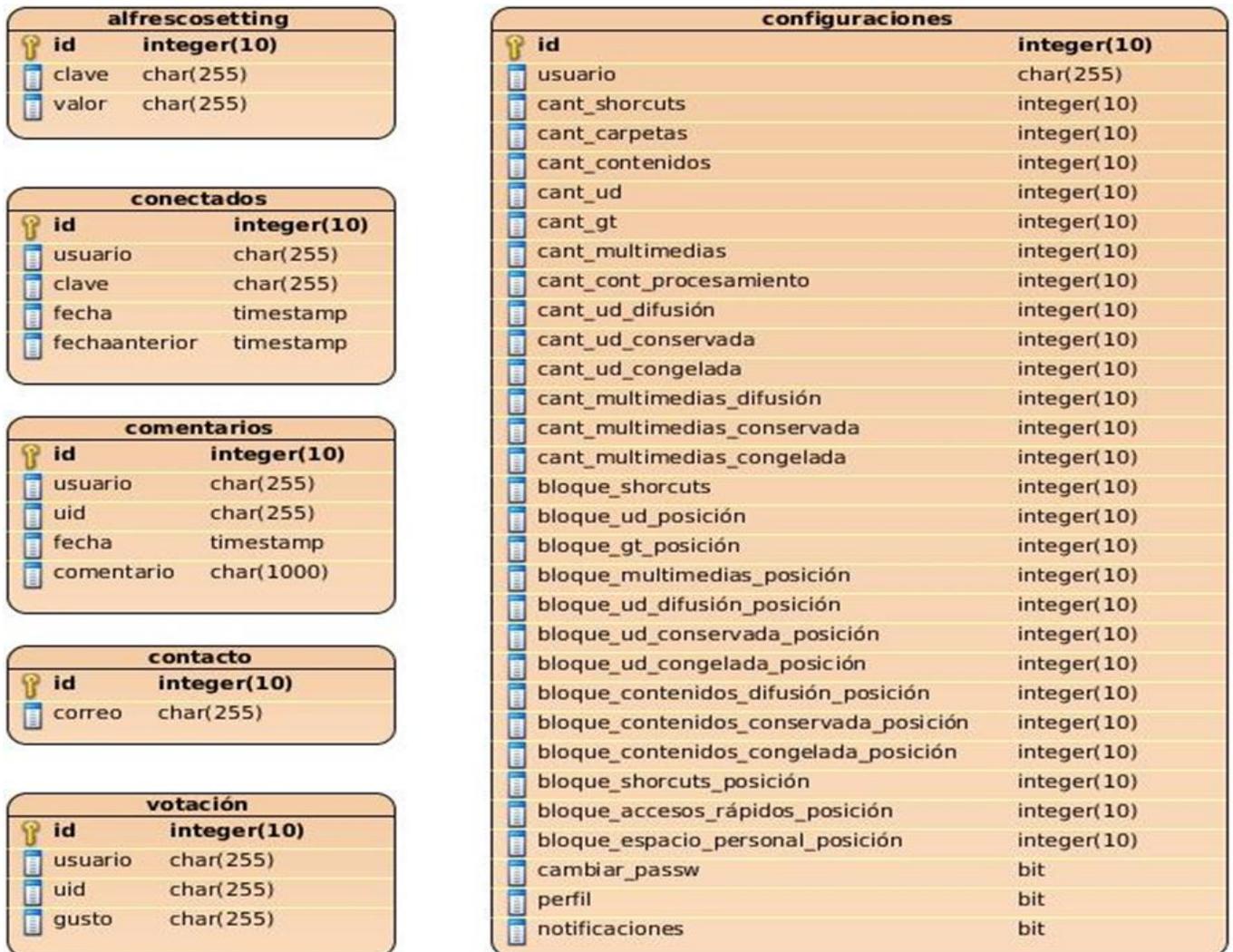


Figura 6. Modelo Entidad - Relación.

A continuación se ofrece una breve descripción de las tablas representadas en el modelo Entidad - Relación:

- **Alfrescosetting:** Contiene las variables necesarias para la conexión con el núcleo del Alfresco así como otras variables de tipo global.
- **Conectados:** Es utilizada para mantener el registro de usuarios conectados a la aplicación lo cual permite la implementación del requisito no funcional de sesión única.
- **Comentarios:** Mantiene el registro de los comentarios de los usuarios.
- **Contacto:** Almacena las direcciones de correos electrónicos a los que se envían los mensajes, quejas y sugerencias de los usuarios.
- **Votación:** Mantiene el registro de los votos de los usuarios asociados a los contenidos.

- **Configuraciones:** Es utilizada para registrar los datos asociados a la configuración personal de cada usuario en la aplicación.

2.6. Modelo de Despliegue

El Modelo de Despliegue se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos y consiste en:

- **Nodos:** elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos
- **Dispositivos:** nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela
- **Conectores:** expresa el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

El diagrama de despliegue del sistema TeVeo Plus V1.0 se muestra en la Figura 7.

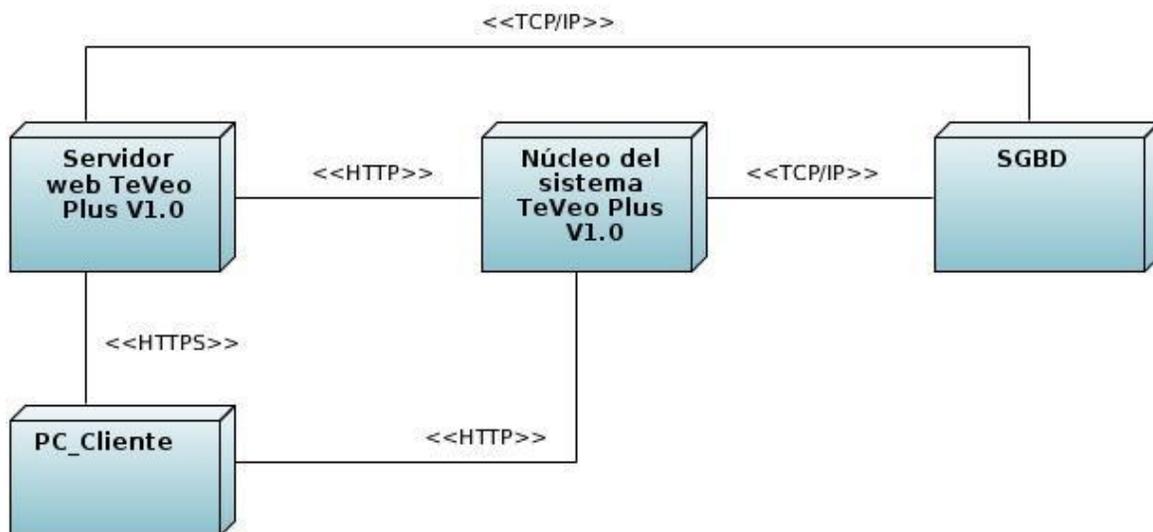


Figura 7. Diagrama de despliegue.

El despliegue de la aplicación está compuesto por:

- Una PC⁶⁷ del lado del cliente, que accederá al servidor web TeVeo Plus V1.0 mediante un navegador web, a través del protocolo HTTPS. Además tendrá acceso al núcleo del sistema a través el protocolo HTTP, puesto que el servidor web del Alfresco está configurado para usar este protocolo.

⁶⁷ Computadora Personal.

- El servidor web TeVeo Plus V1.0 que podrá comunicarse con el núcleo del sistema a través el protocolo HTTP y consumir los servicios web que este brinda.
- Un servidor donde se encuentra el Sistema Gestor de Base de Datos que contendrá las dos bases de datos necesarias para la aplicación. A este servidor podrán conectarse tanto el núcleo del sistema como la interfaz web mediante el protocolo TCP/IP⁶⁸.

2.7. Conclusiones parciales

La propuesta de Aplicación web para el Sistema de Gestión Documental de Audio y Vídeos TeVeo Plus V1.0, integra componentes del núcleo del propio sistema con una serie de elementos de usabilidad, que lo convierten en una solución útil para resolver problemas de gestión de audio y vídeos. En el nuevo sistema el usuario será capaz de desenvolverse de una forma fácil y amena.

El diseño de las funcionalidades solicitadas por el cliente permitió profundizar en el problema a resolver, así como representar una vista interna del sistema en la que se refinó los requisitos y se estructuraron en base a clases y paquetes. Así también el reconocimiento y definición de patrones arquitectónicos y de diseño establecieron las bases para fomentar la reutilización y las buenas prácticas de programación entre los desarrolladores durante la implementación del sistema.

⁶⁸ Modelo de descripción de protocolos de red.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA APLICACIÓN

Una de las etapas decisivas en el desarrollo de software es la de implementación, es en este período donde se define y organiza el código. Entre los objetivos principales destaca el de implementar los elementos de diseño en términos de los elementos de implementación (archivos de origen, binarios, programas ejecutables y otros).

Otra de las etapas importantes del ciclo de desarrollo de un software es la fase de pruebas, pues con esta se busca corregir y perfeccionar el producto desde sus inicios hasta la entrega del mismo al cliente, logrando que el software cumpla exitosamente con todas sus funcionalidades y esté libre de defectos.

3.1. Modelo de componentes

Los diagramas de componentes modelan la vista estática de un sistema. Se representa como un grafo de componentes de software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes, por lo cual cada diagrama describe un apartado del sistema. En la Figura 8 se ilustra el diagrama de componentes del Sistema de Gestión Documental de Audio y Vídeos Digitales Te Veo PlusV1.0.

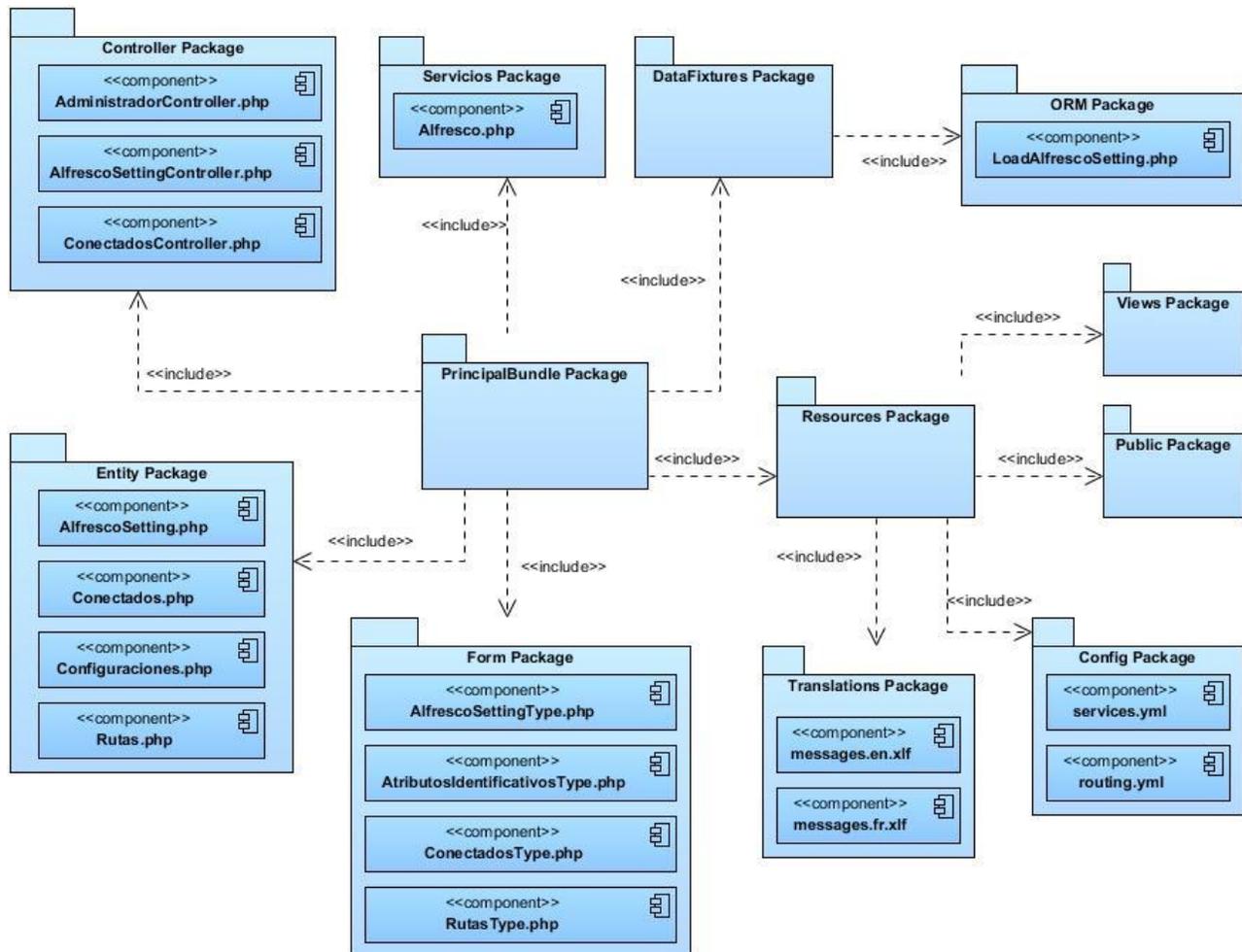


Figura 8. Diagrama de componentes de TeVeo V1.0.

Para el correcto funcionamiento de la aplicación se definen varios paquetes en los cuales se dividen la lógica, la vista, la modelación de los datos, la comunicación con el núcleo, las configuraciones del sistema y los elementos públicos.

- En el paquete **PrincipalBundle** se agrupan todos los componentes, dándole una estructura organizativa al producto.
- El paquete **Controller** está formado por las clases controladoras de la aplicación.
- El paquete **Servicios** contiene los componentes encargados de la comunicación con el núcleo del sistema.
- El paquete **DataFixtures** contiene el paquete ORM.
- El paquete **ORM** contiene los componentes con los datos de pruebas que se insertan en la aplicación
- El paquete **Resources** es el encargado de agrupar los paquetes View, Public, Config y Translations.

- El paquete **View** está conformado por las plantillas HTML de la aplicación, las cuales serán las encargadas de mostrar la información al usuario.
- El paquete **Public** contiene todos los componentes que posibilitan que la interfaz sea amigable, como son las imágenes, los íconos, los CSS, y los archivos JavaScript.
- El paquete **Config** almacena los archivos de configuración del sistema.
- El paquete **Translations** almacena los archivos de traducción del sistema a diferentes idiomas.
- El paquete **Form** almacena los formularios de la aplicación.
- El paquete **Entity** almacena las clases entidades del modelo de datos del sistema.

3.2. Código fuente

Estándar de codificación

Los estándares de código resultan importantes en cualquier proyecto de desarrollo, pero son especialmente importantes cuando muchos desarrolladores trabajan en el mismo proyecto. Los estándares de código ayudan a asegurar que el código tenga una alta calidad, menos errores, y pueda ser mantenido fácilmente.

Las especificaciones del estándar de codificación empleado en esta solución se encuentran registradas en el documento establecido por la metodología OpenUP para la documentación del software (Ver Anexo 3). Entre otros aspectos se alude a las variaciones en las convenciones o adiciones necesarias en el código del lenguaje PHP. Los principales aspectos incluidos en dicho estándar se relacionan a continuación:

- Formato de archivos PHP para la indentación, tamaño máximo de línea, final de línea.
- Convenciones de nombres empleados en clases, clases abstractas, interfaces, nombres de archivo, funciones y métodos, variables, constantes.
- Estilo de código en la demarcación de código PHP, cadenas de caracteres, sustitución de variables, concatenación de cadenas, *arrays*, clases, funciones y métodos, sentencias de control, documentación integrada, formato de documentación, archivos, clases, funciones.

3.3. Pantallas principales de la aplicación

La interfaz de inicio de cualquier aplicación es el elemento más importante, a partir de la impresión que cause al usuario desde el primer momento tendrá la capacidad de definir cuan agradable será la experiencia con la aplicación. En la Figura 9 se ilustra la página de inicio de la aplicación para el usuario colaborador.

Esta interfaz está compuesta por un buscador en la parte superior, seguido de un *banner* con acceso a la ayuda, formulario de contacto y canal RSS. El área de trabajo de esta vista está dividida en cinco bloques:

1. **Unidades Documentales:** contiene las unidades documentales creadas por el usuario.
2. **Archivos Multimedia:** contiene los archivos de audio y vídeos digitales añadidos por el usuario.
3. **Acceso Rápido:** contiene íconos de acceso rápido a las operaciones Configuración, Cambio de contraseña, Visualizar perfil y Notificaciones.
4. **Accesos Directos:** contiene los accesos directos creados por el usuario a las multimedias.
5. **Espacio Personal:** contiene las carpetas y multimedias que el usuario almacena para procesarlas posteriormente.



Figura 9. Vista de la aplicación para el usuario colaborador.

El resto de las pantallas principales de la aplicación se ilustran en el Anexo 4.

3.4. Validación del sistema

Las pruebas de validación en la ingeniería de software son el proceso de revisión que verifica que el sistema producido cumple con las especificaciones y que logra su objetivo. Es una parte del proceso de pruebas de software de un proyecto. La validación es el proceso de comprobar que lo que se ha especificado es lo que el usuario realmente solicitó. Se evalúa el sistema o parte de este, durante o al final del desarrollo, para determinar si satisface los requisitos iniciales.

3.4.1. Pruebas funcionales

Se denominan pruebas funcionales a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados cumplan con las funciones específicas para los cuales han sido creados. A este tipo de pruebas se les denomina también pruebas de comportamiento y para realizarlas se emplea el método de caja negra, donde los probadores o analistas de pruebas no enfocan su atención en cómo se generan las respuestas del sistema, sino en el funcionamiento de la interfaz del sistema. Básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas. (37)

El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. A continuación se muestran la propuesta de diseño de casos de prueba basados en casos de uso a realizar a la aplicación TeVeo Plus V1.0.

Diseño de casos de prueba basados en casos de uso

Caso de uso: Autenticar usuario

Condiciones de ejecución:

- El usuario debe entrar al navegador web y conectarse al sistema mediante la dirección <http://teveoplus.uci.cu/>
- Esta funcionalidad está habilitada para todos los usuarios del sistema.

Escenario	Descripción	Variable 1	Variable 2	Respuesta del sistema	Flujo central
Autenticar usuario.	Entrada de los datos de autenticación	V ⁶⁹	V	Entrar al espacio personal	- Llenar correctamente los campos del

⁶⁹ Debe ser interpretado como válido.

	correctamente.			del usuario autenticado.	formulario de autenticación. - Luego presionar el botón autenticar.
Mostrar mensaje de error de usuario incorrecto.	El campo de usuario es incorrecto.	I ⁷⁰	V	Mensaje de error que especifica “Verifique sus datos.”	- Llenar los campos del formulario de autenticación proporcionando un usuario incorrecto. - Luego presionar el botón autenticar.
Mostrar mensaje de error de contraseña incorrecta.	El campo de contraseña es incorrecto.	V	I	Mensaje de error que especifica “Verifique sus datos.”	- Llenar los campos del formulario de autenticación proporcionando una contraseña incorrecta. - Luego presionar el botón autenticar.
Mostrar mensaje de error de campo vacío.	El campo Usuario está vacío.	NA ⁷¹	V	Mensaje de error “Este campo no puede estar vacío.”	- Llenar solo el campo de Usuario del formulario de autenticación. - Luego presionar el botón autenticar.
Mostrar mensaje de error de campo vacío.	El campo Usuario está vacío.	V	NA	Mensaje de error “Este campo no puede estar vacío.”	- Llenar solo el campo de Contraseña del formulario de autenticación. - Luego presionar el botón autenticar.

Tabla 5. Diseño de caso de prueba del caso de uso Autenticar Usuario.

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Usuario	Campo de texto.	No	Máximo 20 caracteres. Mínimo 3 caracteres.

⁷⁰ Debe ser interpretado como inválido.

⁷¹ Debe ser interpretado como que no es necesario ingresar un valor de dato, ya que es irrelevante.

				No admite números. No admite caracteres extraños (@,#,~,,\$,%,&,/,,-)
2	Contraseña	Campo de contraseña.	No	Máximo 20 caracteres. Mínimo 5 caracteres.

Tabla 6. Variables empleadas en el diseño de caso de prueba Autenticar Usuario.

Los diseños de casos de prueba asociados a los casos de uso Adicionar contenido y Eliminar contenido pueden ser consultados en el Anexo 5.

Resultados de las pruebas funcionales

Para la validación de los requisitos funcionales se realizaron 3 iteraciones de pruebas. A continuación se muestran los resultados obtenidos en cada una de las iteraciones de pruebas realizadas a la aplicación TeVeo Plus V1.0 (Ver Tabla 7).

No conformidades	Primera iteración	Segunda iteración	Tercera iteración
Detectadas	24	19	8
Resueltas	19	17	8
Pendientes	5	2	0

Tabla 7. Comportamiento de las no conformidades por iteraciones.

En la Figura 10 se puede observar el comportamiento de las no conformidades detectadas en cada una de las iteraciones de pruebas realizadas a la aplicación.

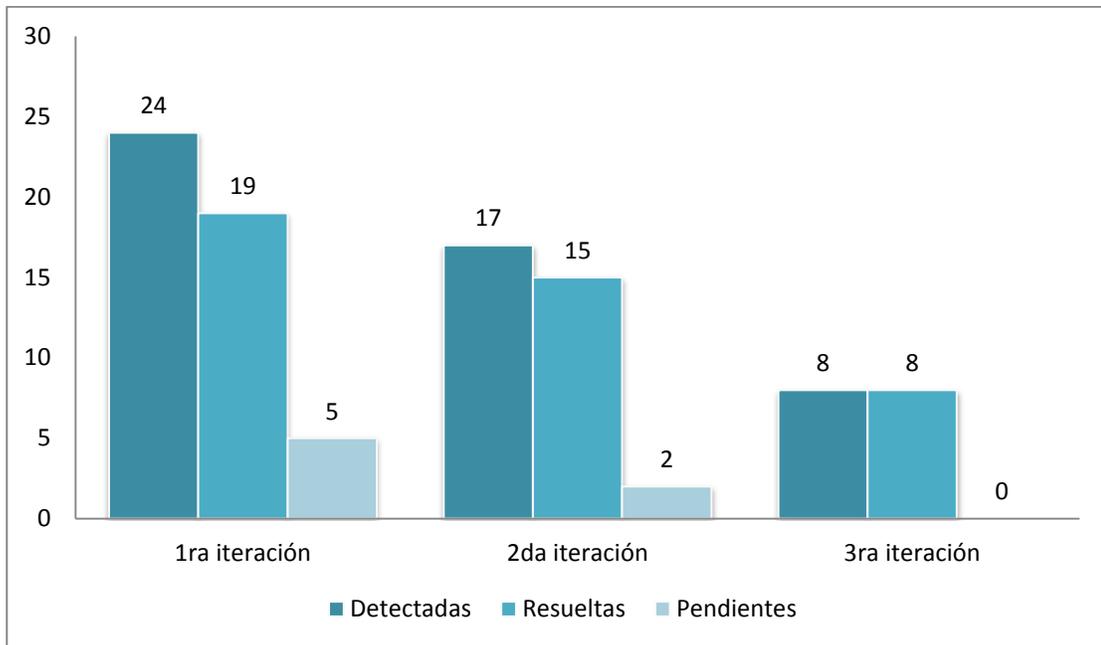


Figura 10. Comportamiento de las no conformidades en las pruebas funcionales.

Entre las no conformidades detectadas durante el proceso de pruebas se destacan las siguientes:

- No se muestran mensajes de confirmación cuando se mueve un contenido de un espacio a otro.
- No se guardan los resultados de las búsquedas.
- En algunos formularios aún existen campos que deberían ser autocompletados.
- No se brinda suficiente información en la ayuda online.

3.4.2. Pruebas de seguridad

Las pruebas de seguridad garantizan que los usuarios estén restringidos a funciones específicas o que su acceso esté limitado únicamente a los datos que están autorizados a acceder. Sólo aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar las funciones disponibles. Su objetivo fundamental es comprobar los niveles de seguridad lógica del sistema.

Los especialistas del grupo de Seguridad del DEPSW⁷² de la UCI han establecido 3 niveles para realizar las pruebas de seguridad. Para evaluar la seguridad de las aplicaciones en un primer nivel (nivel 1) se definió una lista de chequeo que cuenta con 15 indicadores separados en 4 tipos de pruebas accesibles a los 3 niveles.

⁷² Departamento de software.

La aplicación de la evaluación mediante dicha lista arrojó, para los 4 tipos de pruebas, los siguientes resultados:

1. **Pruebas de Autorización:** Ningún usuario estándar puede modificar sus privilegios ni los de otro usuario en la aplicación.
2. **Pruebas de Gestión de Sesiones:** Se puede acceder a la aplicación copiando la URL después de estar autenticado, cerrar el navegador y volver a abrirlo, debido a que el tiempo de vida de la sesión es de 10 horas. No se puede acceder al cerrar la sesión de un usuario y dar clic en el botón “Atrás” del navegador.
3. **Validación de Datos:** Se enmascaran datos confiables cuando se visualizan en la aplicación. Solamente se permiten contraseñas alfanuméricas, que incluyan caracteres especiales y que tengan seis caracteres mínimos de longitud. La funcionalidad de cambio de contraseña únicamente se permite a usuarios autenticados validando la antigua contraseña, la nueva contraseña y la respuesta a la pregunta de seguridad. El sistema no muestra mensajes indebidos al colocar en la barra de dirección o en campos de entrada los caracteres: (‘, &, +, -, /)
4. **Comprobación del Sistema de Autenticación:** Los mensajes de error para distintas combinaciones de autenticación muestran la misma información. Los tiempos de respuestas usuario correcto - contraseña incorrecta y usuario - contraseña incorrecta son los mismos. El sistema protege el envío de los datos mediante protocolo seguro (HTTPS) y se bloquea la sesión del usuario después de 15 min estando sin usar la misma. Pero aún no se bloquea la cuenta del usuario después de un número N de intento de autenticación fallidos.

Esta lista de chequeo permitió recoger los puntos eficientes y los ineficientes que tienen los elementos chequeados, así como verificar que el grado de seguridad de la aplicación es adecuado para la protección de la información.

Para evaluar la seguridad de la aplicación en un segundo nivel (nivel 2) se hizo uso de la herramienta Acunetix WVS 8. Durante el escaneo realizado por la herramienta fueron detectadas solo 6 no conformidades, de ellas 2 de prioridad baja y 4 informativas. Entre las no conformidades de prioridad baja se encontraron: ficheros del servidor Apache vulnerables a ataques de fuerza bruta y *cookies* de sesión sin el indicador de seguridad. El resto deja al descubierto información acerca de una página de error que contiene la versión del servidor web y una lista de los módulos habilitados en este servidor.

A pesar de las vulnerabilidades detectadas, el sistema implementado cumple casi la totalidad de los indicadores establecidos para los niveles 1 y 2 de seguridad implantados en la UCI. Por tanto TeVeo Plus V1.0 no es considerada una aplicación vulnerable y el sistema de seguridad actual puede ser suficiente para preservar la confidencialidad, integridad y disponibilidad de los archivos multimedia.

3.4.3. Pruebas de carga y estrés

Las pruebas de carga y estrés se llevaron a cabo empleando JMeter, una herramienta automatizada existente en la universidad. Apache JMeter: es una aplicación java que permite definir comportamientos para casos de pruebas y medir su rendimiento. Esta herramienta permite probar el sistema trabajando bajo condiciones de carga y bajo fallo.

Estas pruebas fueron realizadas para un usuario con privilegios mínimos debido a que la implementación del requisito no funcional de sesión única imposibilita realizarlas a otro usuario.

Las pruebas de carga fueron realizadas para 120 usuarios conectándose una vez en intervalos de tiempo de un segundo. Dichas pruebas arrojaron los siguientes resultados:

- El servidor es capaz de responder a 40 peticiones por segundo, con una media de 9946 páginas cargadas satisfactoriamente. El menor tiempo que demoró en cargar una página fue de 0.23 segundos. El máximo de tiempo que demoró en cargar una página fue de 3.1 segundos. El error en las respuestas fue de 0%.

Con esta prueba se pudo verificar que el sistema satisface los requisitos de rendimiento para la cantidad de usuarios especificados por el cliente, accediendo de forma concurrente a los servicios del programa.

Las pruebas de estrés fueron realizadas para 450 usuarios conectándose una vez en intervalos de tiempo de un segundo. Dichas pruebas arrojaron los siguientes resultados:

- El servidor es capaz de responder a 20 peticiones por segundo, con una media de 7780 páginas cargadas satisfactoriamente. El menor tiempo que demoró en cargar una página fue de 0.3 segundos. El máximo de tiempo que demoró en cargar una página fue de 5.4 segundos. El error en las respuestas fue de 2.22%.

Con esta prueba se sometió a la aplicación a una carga elevada para comprobar el límite que puede soportar y se evidenció que hubo un aumento en la demora de las respuestas, conjuntamente comenzaron a aparecer errores. Por tanto se recomienda que no sea accedida por un número mayor que 450 usuarios concurrentes.

3.4.4. Pruebas de usabilidad

Los métodos de inspección de la usabilidad de una aplicación web son aquellos realizados por el experto en usabilidad, y que se basan en el recorrido y análisis del sitio identificando errores y problemas de

diseño. Las pruebas de usabilidad garantizan el aumento de la medida en que el producto puede ser usado para conseguir objetivos específicos con eficiencia y la satisfacción del usuario.

Los especialistas del grupo de Calisoft⁷³ de la UCI han establecido una lista de chequeo que cuenta con varios puntos, los mismos están agrupados en 12 categorías. La evaluación de los indicadores contenidos en dicha lista arrojó, para las 12 categorías, los siguientes resultados:

Categoría de Indicador	Indicadores	Defectuosos	Mejorados
Visibilidad del sistema	7	3	3
Adecuación del sistema	18	12	11
Control y libertad de usuarios	13	6	4
Consistencia y estándares	17	4	2
Prevención de errores	4	3	3
Reconocimiento	9	5	5
Flexibilidad y eficiencia de uso	7	4	4
Estética	18	5	5
Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores	5	5	4
Ayuda y documentación	10	10	10
Accesibilidad	43	19	11
Comprobaciones técnicas	5	2	1
TOTAL	156	78	63

Tabla 8. Estadísticas obtenidas a partir de la lista de chequeo de usabilidad.

Para el total de 156 indicadores verificados el Sistema de Gestión Documental de Medias para la Prensa (SGDMP) presenta un 50% de usabilidad, luego de implementar la solución TeVeo la usabilidad fue incrementada a un 90,4%, lo cual representa un incremento del 40,4%. La representación gráfica de estos datos puede ser observada en los gráficos de las Figuras 11 y 12.

⁷³ Centro Nacional de Calidad del Software.

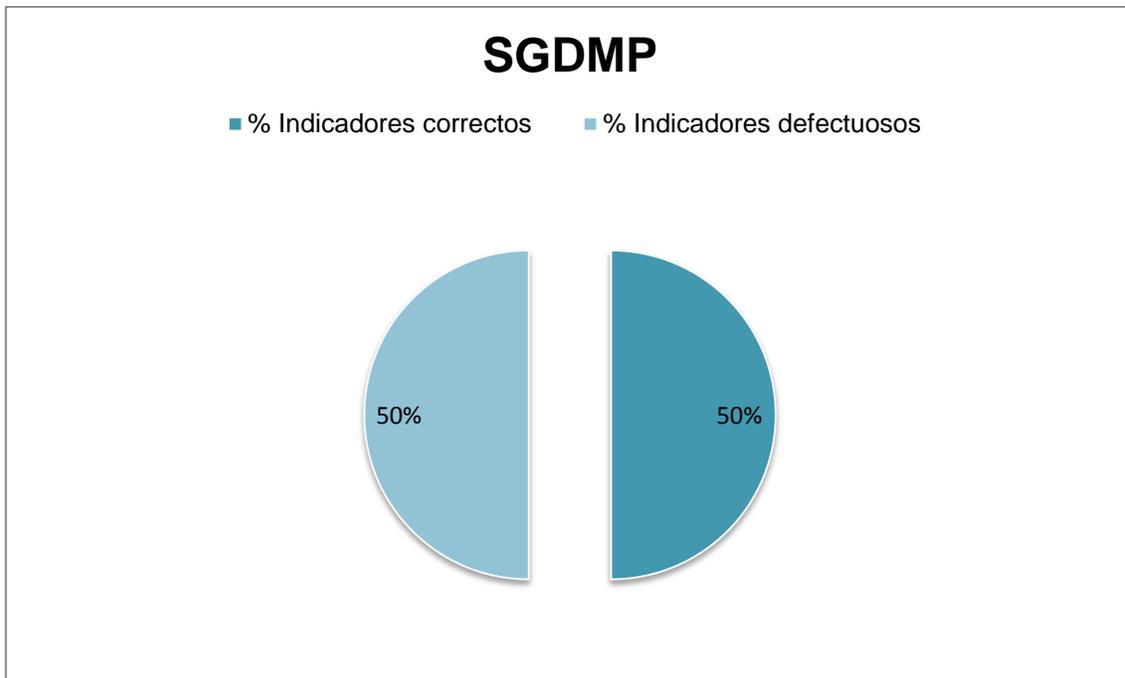


Figura 11. Comportamiento de los indicadores de usabilidad en el SGDMP.

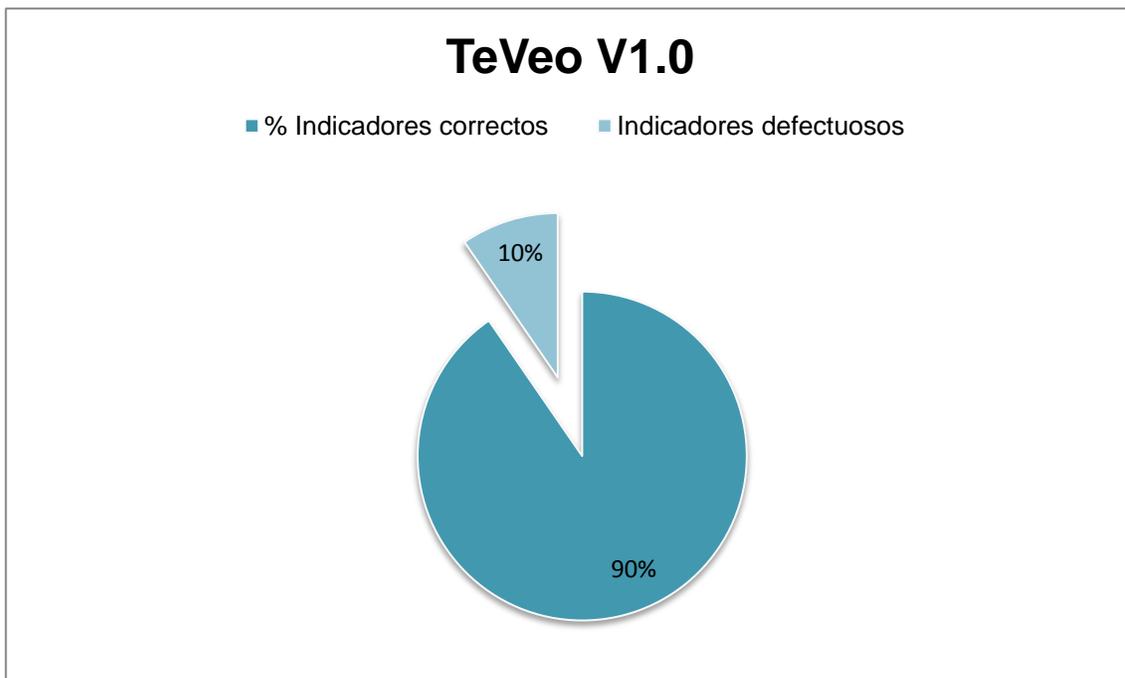


Figura 12. Comportamiento de los indicadores de usabilidad en TeVeo Plus V1.0.

3.5. Conclusiones parciales

La implementación del sistema se llevó a cabo aprovechando al máximo las propiedades de reutilización gracias a la selección de tecnologías modernas, a la utilización de patrones y al modelado de una vista

estática del sistema en el diagrama de componentes. De igual modo los estándares de código permitieron adoptar una estructura homogénea que facilita la comunicación y asegura la calidad, menos errores y fácil mantenimiento.

La detección temprana de errores, mediante la aplicación de pruebas de funcionalidad, seguridad, carga y estrés y usabilidad, permitió validar que: el sistema trabaja como fue diseñado, que los requisitos no fueron violados, que la implementación fue correcta, que se cumplieron los principios básicos de la seguridad y que la usabilidad fue incrementada en un porcentaje considerable.

CONCLUSIONES

Al término de la fundamentación teórica que sustentó el presente trabajo y del desarrollo y validación del sistema TeVeo Plus V1.0, se arriba a las siguientes conclusiones:

- Durante la evaluación de las tendencias de aplicaciones similares fueron identificados sistemas con un alto nivel de aceptación, que aunque no satisfacen ni las expectativas del proyecto para realizar un uso correcto de la gestión documental, ni las necesidades y restricciones del cliente en cuanto a la representación y manejo de audio y vídeos digitales, permitieron adquirir nuevos elementos que tributan al mejoramiento del sistema.
- Con la modelación de esquemas y creación de artefactos, se generó una propuesta de solución que permitió obtener una visión del adecuado funcionamiento del sistema, que guió el desarrollo de la aplicación.
- La aplicación web TeVeo PlusV1.0 proporciona a los usuarios la posibilidad de almacenar, consultar y buscar los recursos multimedia, de forma rápida y sencilla, contribuyendo de esta forma a optimizar el trabajo de los profesionales del sector periodístico, investigativo y otras instituciones que hagan uso de estos contenidos.
- Fue comprobada la efectividad de la solución propuesta a partir de los resultados satisfactorios obtenidos en las evaluaciones internas realizadas por el proyecto, pues la totalidad de las no conformidades detectadas durante las iteraciones de pruebas definidas fueron resueltas.

RECOMENDACIONES

Una vez concluida la investigación y el desarrollo del sistema se recomienda:

- Incluirle nuevas funcionalidades a la Aplicación web TeVeo Plus V1.0 para complementar la gestión documental de audio y vídeos digitales, tales como:
 - Auditar contenidos
 - Exportar resultados de búsquedas
- Implementar las funcionalidades editar rutas y gestionar taxonomías en espacio de administración.

REFERENCIAS BIBLIOGRÁFICAS

1. **Campillo Torres, Irima.** *Sistema de Gestión Integral de Documentos de archivo para empresas de la construcción del territorio de Camagüey.* Granada : Universidad de Granada, 2010. 299.
2. **AIIM.** AIIM. [En línea] 2011. [Citado el: 24 de 10 de 2012.] <http://www.aiim.org/What-is-ECM-Enterprise-Content-Management>.
3. **De la Cruz Leyva, Yusmila.** *Sistema de Gestión Documental. Módulo Fototeca para las Fotografías de la Prensa.* La Habana : s.n., 2011. 83.
4. **CIDI.** *Gestión Documental de Medias Núcleo del Sistema de Gestión Documental de Medias V1.0.* La Habana : s.n., 2011.
5. **Fugueras, Alberch.** *Los archivos entre la memoria histórica y sociedad del conocimiento.* Barcelona : UOC, 2003.
6. **Ponjuan, Dante.** Gestión documental, gestión de información y gestión del conocimiento: evolución y sinergias. [En línea] 2005. [Citado el: 2012 de 11 de 16.] <http://www.cinfu.cu/Userfiles/file/Cinfo/CINFO/diciembre2005/Comunicación.doc>.
7. **Mugica, Mena.** *Gestión documental y organización de archivos.* La Habana : Félix Varela, 2005.
8. **Cuauhtémoc Robles, Cairo.** *La Mediateca, una obra de la informática del nuevo siglo.* 2001.
9. **Real Academia de la Lengua Española.** Diccionario Usual. *Real Academia Española.* [En línea] 2012. [Citado el: 2 de 11 de 2012.] <http://www.rae.es/rae.htm>.
10. —. Diccionario Usual. *Real Academia Española.* [En línea] 2012. [Citado el: 3 de 11 de 2012.] <http://www.rae.es/rae.htm>.
11. **Binovo.** Binovo. *Binovo.* [En línea] 2012. [Citado el: 21 de 11 de 2012.] http://www.binovo.es/index.php?option=com_content&view=article&id=110&Itemid=89.
12. **Hassan, Yusef.** Introducción a la Usabilidad. *Introducción a la Usabilidad.* [En línea] 01 de 11 de 2002. [Citado el: 2 de 4 de 2013.] http://www.nosolousabilidad.com/introduccion_usabilidad.html. 1886-8592.
13. *Architecting for usability: a survey.* *Journal of Systems and Software.* **Folmer, E. y Bosch, J.** 2004, Vol. 70. 1-2.
14. **9241-11, ISO.** *Ergonomic requirements for office work with visual display terminals - Part 11 Guidance on usability.* 1998.

REFERENCIAS BIBLIOGRÁFICAS

15. **Hassan Montero, Y. y Martín Fernández, F.J.** Qué es la Accesibilidad Web. *Qué es la Accesibilidad Web*. [En línea] 16 de 5 de 2003. [Citado el: 2 de 4 de 2013.] <http://www.nosolousabilidad.com/articulos/accesibilidad.htm>.
16. **Hassan, Yusef y Martín, Francisco.** Qué es la Accesibilidad Web. *Qué es la Accesibilidad Web*. [En línea] 14 de 07 de 2003. [Citado el: 28 de 3 de 2013.] <http://www.nosolousabilidad.com/accesibilidad.html>. 1886-8592.
17. **Henry, Shawn.** Understanding Web Accessibility. *En Constructing Accessible Web Sites*. [En línea] 4 de 2002. [Citado el: 28 de 3 de 2013.] http://www.macromedia.com/macromedia/accessibility/pub/acc_sites_chap01.pdf. 1904151000.
18. **Nielsen, Jakob.** Beyond Accessibility: Treating People with Disabilities as People. [En línea] 11 de 11 de 2001. [Citado el: 2 de 4 de 2013.] <http://www.useit.com/alertbox/20011111.html>.
19. **YouTube.** YouTube. *YouTube*. [En línea] 2 de 2005. [Citado el: 24 de 10 de 2012.] <http://www.youtube.com>.
20. **Vimeo.** Vimeo. *Vimeo*. [En línea] 2004. [Citado el: 21 de 10 de 2012.] <http://www.vimeo.com>.
21. **TeVeo.** TeVeo. *TeVeo*. [En línea] [Citado el: 25 de 10 de 2012.] <http://teveo.icrt.cu/>.
22. **Hernández, Ruber.** *Primicia, Plataforma de Televisión Informativa*. La Habana : s.n., 2009.
23. **Alfresco.** Alfresco explorer. *alfrescowiki*. [En línea] 2011. [Citado el: 2 de 11 de 2012.] http://wiki.alfresco.com/wiki/Alfresco_Explorer.
24. **Opsoro.** Opsoro. *opsoro*. [En línea] 2011. [Citado el: 4 de 11 de 2012.] <http://www.opsoro.org/>.
25. **iFresco.** iFresco. *ifresco*. [En línea] 2011. [Citado el: 2 de 11 de 2012.] <http://www.ifresco.at/de/>.
26. **Elejalde, Reinier.** Módulo para la creación de modelos de contenido para Alfresco. [En línea] 7 de 2009. [Citado el: 12 de 11 de 2012.] http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2390_09.
27. **Symfony, la guía definitiva.** librosweb.es. [En línea] 2011. [Citado el: 11 de 20 de 2012.] <http://www.librosweb.es/symfony/>.
28. **Alvarez, Miguel Angel .** DesarrolloWeb.com. *manual de jQuery*. [En línea] 4 de 4 de 2012. [Citado el: 24 de 11 de 2012.] www.desarrolloweb.com/manuales/manual-jquery.html.
29. **Bootstrap.** Bootstrap. [En línea] [Citado el: 28 de 11 de 2012.] <http://twitter.github.io/bootstrap/index.html>.
30. **Flumotion.** Flumotion. [En línea] 2012. [Citado el: 25 de 5 de 2013.] <http://www.flumotion.net/features/>.
31. **Martínez, Rafael.** PostgreSQL-es. *Portal en español sobre PostgreSQL*. [En línea] 2009. [Citado el: 28

REFERENCIAS BIBLIOGRÁFICAS

de 11 de 2012.] http://www.postgresql-es.org/sobre_postgresql.

32. **CIDI**. *Gestión Documental de Medias Núcleo del Sistema de Gestión Documental de Medias V1.0*. La Habana : s.n., 2011.

33. **Frederick P. Brooks, Jr.** *The Mythical Man-Month*. s.l. : Addison-Wesley, 1995.

34. **Symfony, la guía definitiva**. Symfony. *Symfony*. [En línea] 2011. [Citado el: 16 de 2 de 2013.] <http://www.librosweb.es/symfony/>.

35. **Larman, Craig**. *Introducción al Análisis y Diseño Orientado a Objetos*. Uruguay : s.n., 2008.

36. **Gamma**. *Design Patterns*. 1995.

37. **Calidad y software**. Calidad y software. *Pruebas funcionales*. [En línea] 2009. [Citado el: 23 de 3 de 2013.] http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.

BIBLIOGRAFÍA

Alfresco Software, Inc. Alfresco. [En línea] 2013. <http://www.alfresco.com/es/>.

Angel, Miguel. desarrolladoresweb.com. *¿Qué es JavaScript?* [En línea] 2011. [Citado el: 24 de 1 de 2013.] <http://www.desarrolloweb.com/articulos/25.php>.

Blanco, Jaime y Bernaus, Albert. *Aprender a crear páginas Web.* [documento] 2006.

Calidad y software. Calidad y software. *Pruebas funcionales.* [En línea] 2009. [Citado el: 25 de 2 de 2013.] http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.

Cei, Ugo y Lucidi, Piergiorgio. *Alfresco 3 Web Services.* [documento] Olton : Packt Publishing Ltd, 2009. 1110810.

Eguíluz, Javier. librosweb. *Introducción a JavaScript.* [En línea] 2009. [Citado el: 2 de 11 de 2012.] <http://www.librosweb.es/javascript>.

Facundo, Héctor. *Sitios Web bajo Linux.* Buenos Aires : MP Ediciones, 2006. ISBN 987-526-074-6.

Fernández Valderrama, Luis David. Gestión Documental. [En línea] 2010. [Citado el: 25 de 12 de 2012.] [http://www.sociedadelainformacion.com/12/Gestion Documental.pdf](http://www.sociedadelainformacion.com/12/Gestion%20Documental.pdf).

Flores, Katherine. Conceptos Básicos Relacionados al Internet. *Internet y sus avances.* [En línea] 2010. [Citado el: 3 de 12 de 2012.] <http://katherineflores.com/2010/09/00002-conceptos-basicos-relacionados-al-Internet.html>.

Silberschatz, Abraham, F. Korth, Henry y Sudarshan. *Fundamentos de bases de datos.* s.l. : S. McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U, 2010.

Huallpara, Marca y Limachi, Quisbert. *Diagrama de Despliegue.* 2009.

John. Manuales de Ayuda. [En línea] 2010. [Citado el: 4 de 12 de 2012.] <http://www.manualesdeayuda.com/manuales/bases-de-datos/postgresql/caracteristicas-de-postgresql-01844.html>.

Martínez, Rafael. Libros sobre PostgreSQL. [En línea] 2009-2011. [Citado el: 29 de 11 de 2012.] <http://www.postgresql.org.es/libros>.

Pecos, Daniel. PostgreSQL vs. MySQL. [En línea] [Citado el: 1 de 12 de 2012.] http://danielpecos.com/docs/mysql_postgres/x15.html.

Redondo Duque, Miguel y Bravo Santos, Crescencio. *Sistemas interactivos y colaborativos en la Web.*

2006.

Shariff, Munwar. *Alfresco Enterprise Content Management Implementation*. s.l. : PACKT, 2010.

Tore, Horacio Aldo. Patrones de diseño de software. *Patrones de diseño de software*. [En línea] 2010. [Citado el: 12 de 2 de 2013.] <http://www.dipneo.com.ar>.

Visual Paradigm. Visual Paradigm Internacional. [En línea] 2005. [Citado el: 21 de 10 de 2012.] <http://www.visual-paradigm.com/product/vpuml/>.

ANEXOS

Anexo 1. Diagramas de proceso.

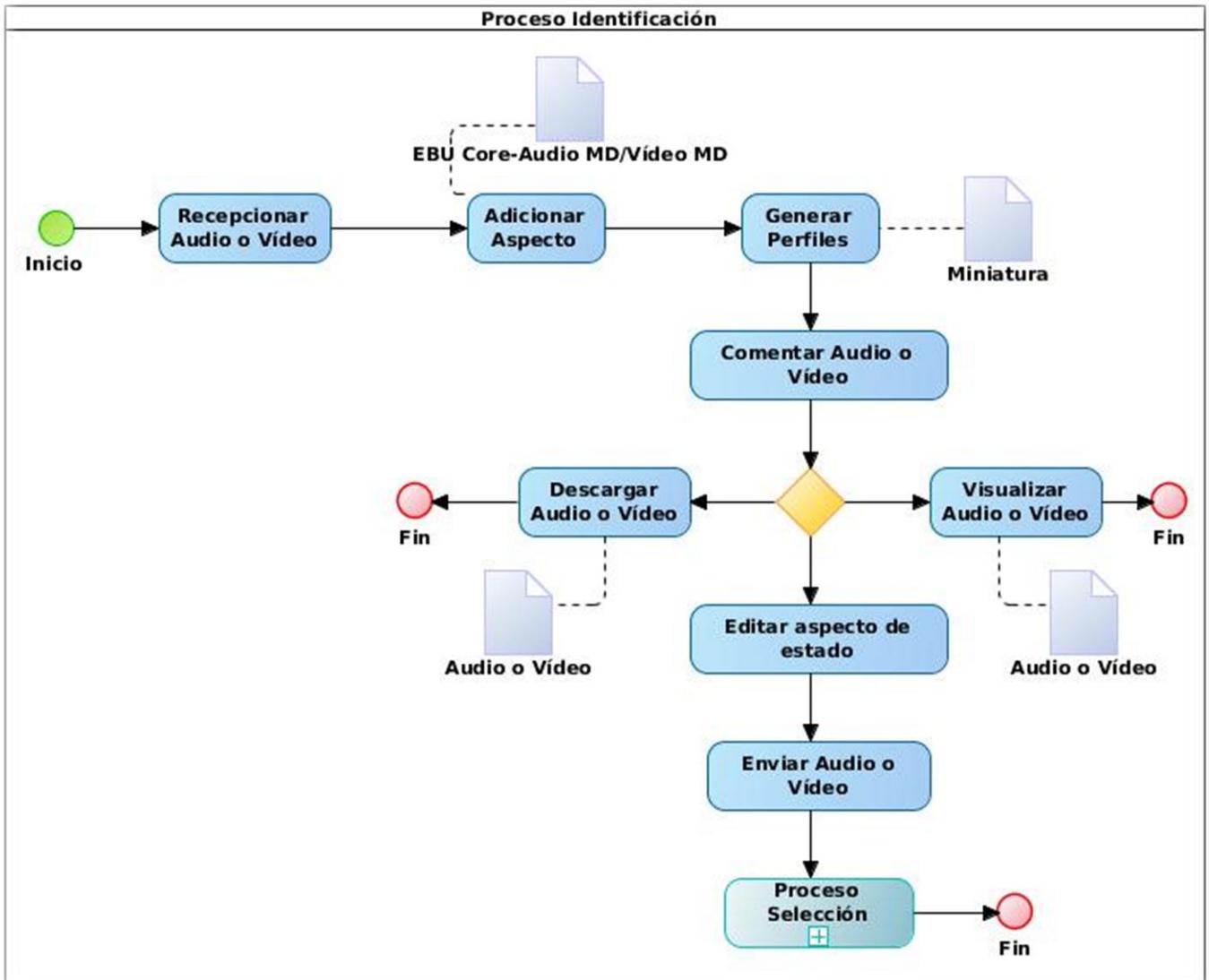


Figura 13. Proceso identificación.

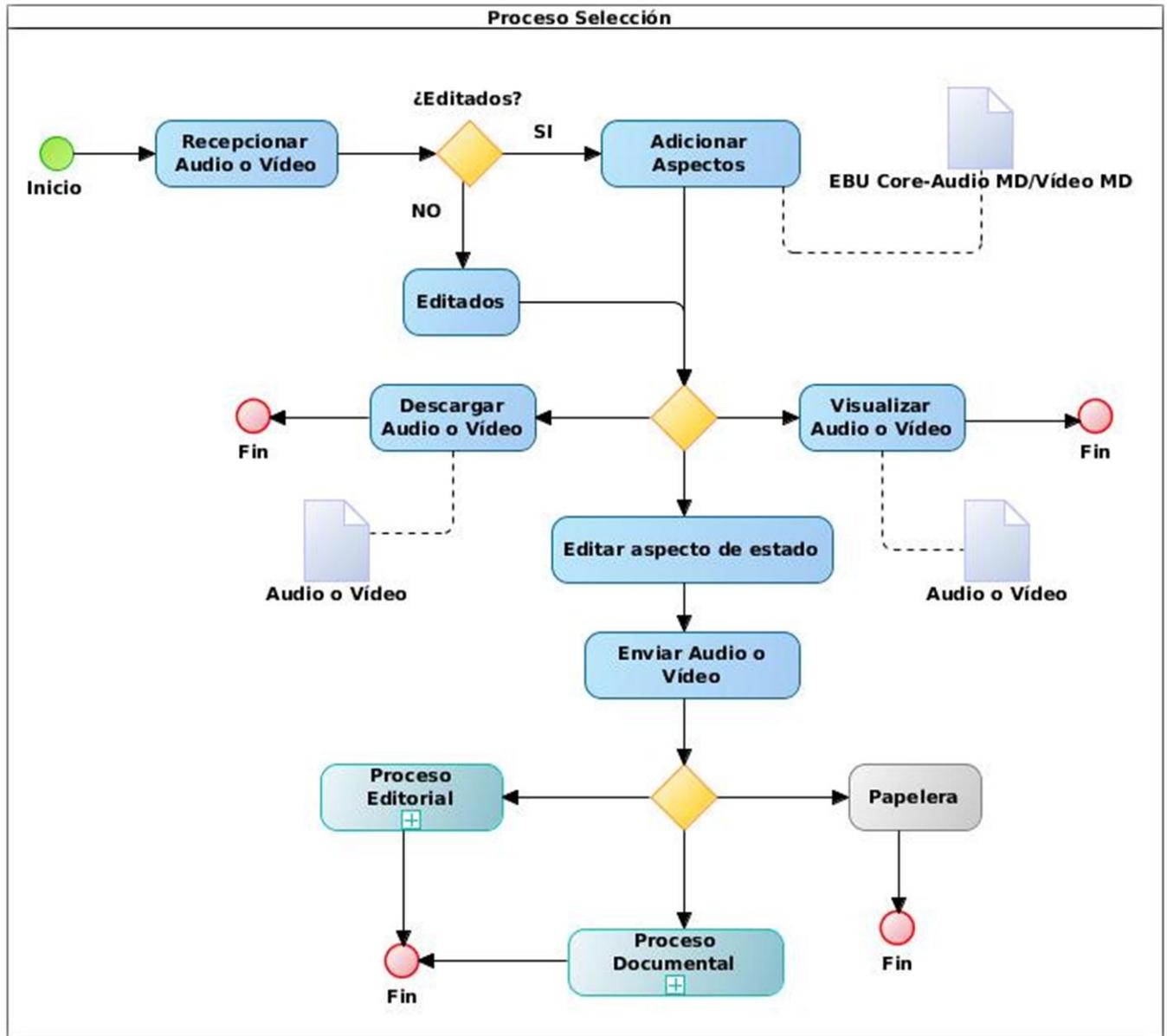


Figura 14. Proceso Selección.

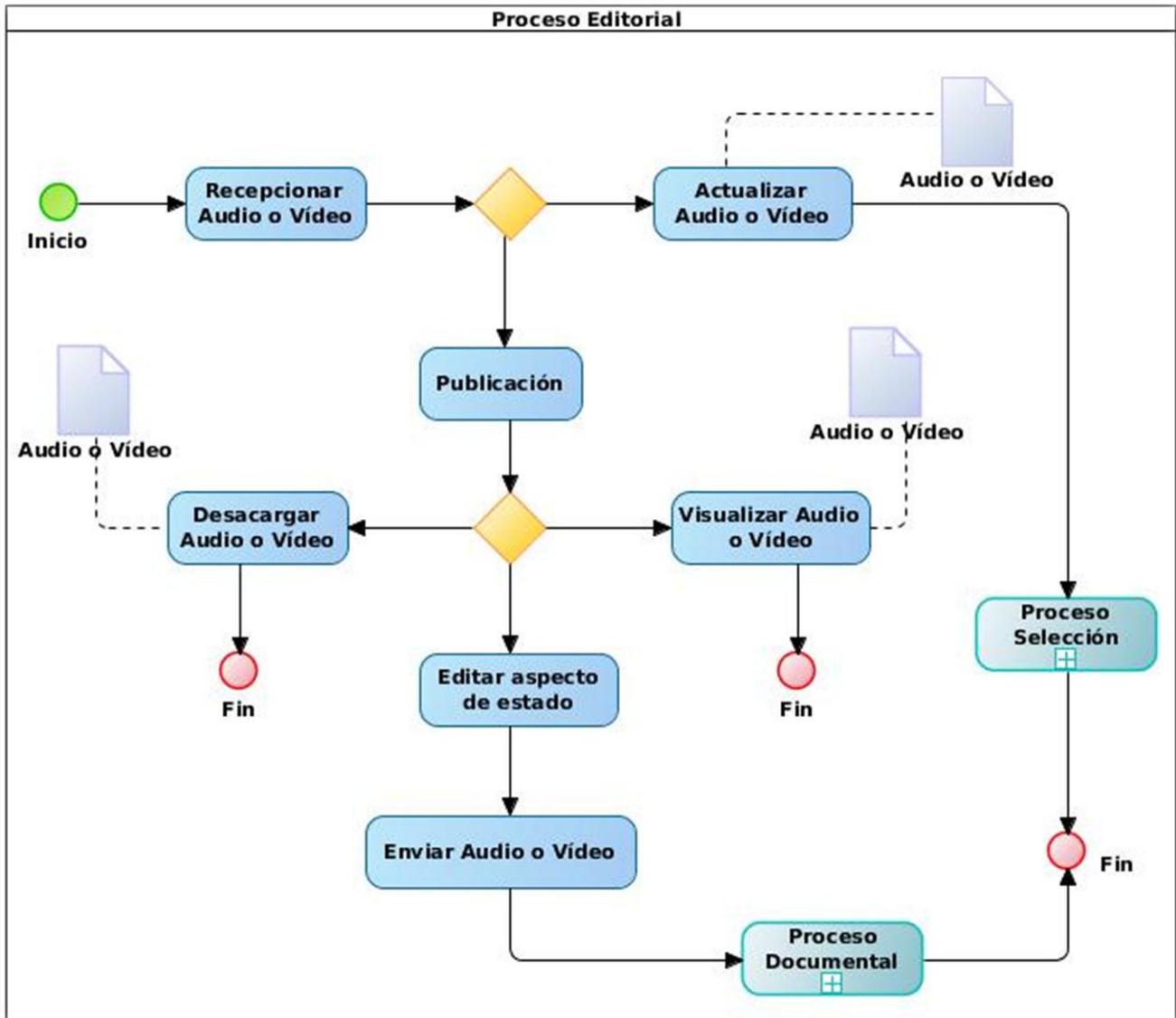


Figura 15. Proceso Editorial.

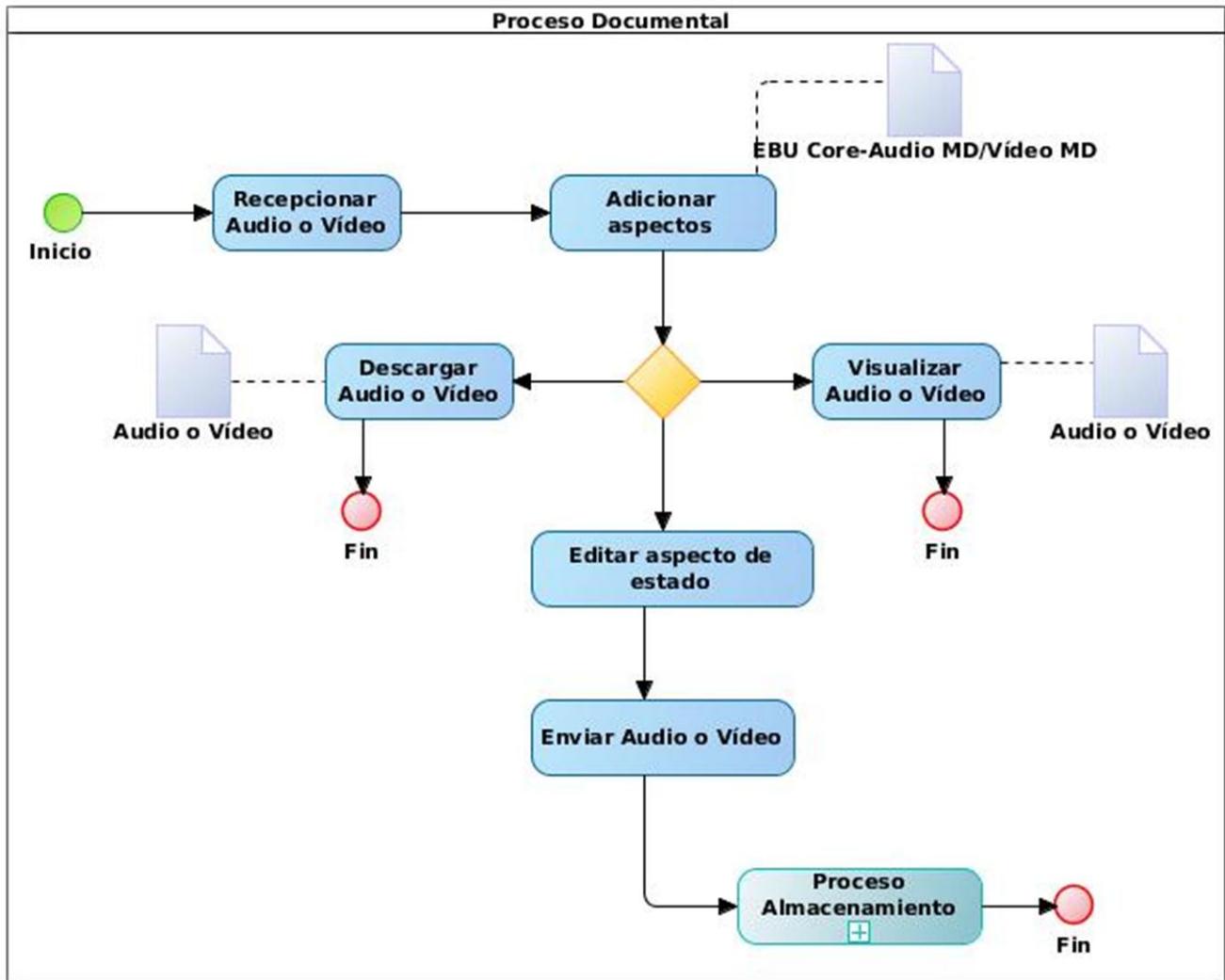


Figura 16. Proceso Documental.

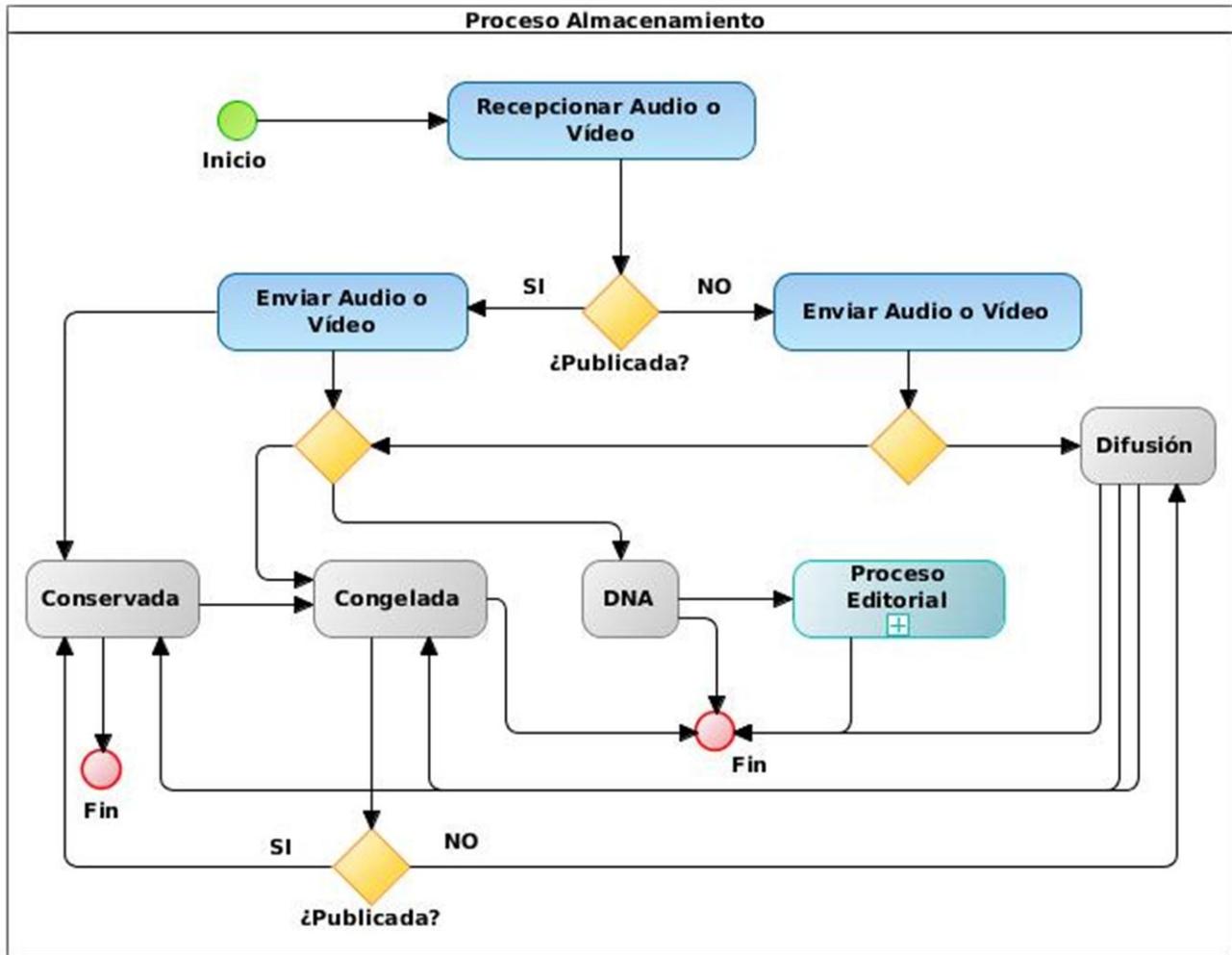


Figura 17. Proceso Almacenamiento.

Anexo 2. Descripción de Casos de Uso

CU_2. Adicionar Contenidos

Objetivo	Adicionar contenido en el sistema.	
Actores	Administrador: (Inicia) Adiciona contenido al sistema. Colaborador:(Inicia) Adiciona contenido al sistema. Gestor:(Inicia) Adiciona contenido al sistema.	
Resumen	El CU permite adicionar un nuevo contenido al sistema.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Usuario del sistema autenticado.	
Postcondiciones	Contenido adicionado al sistema.	
Flujo de eventos		
Flujo básico Adicionar contenidos.		
	Actor	Sistema
1.	Autenticado en el sistema y ubicado en su espacio de trabajo selecciona la opción adicionar contenido.	Muestra una ventana emergente para introducir los datos identificativos.
2.	Rellena los campos solicitados.	Valida que el archivo cumpla con las especificaciones establecidas. Valida que los datos estén correctos y no existan en el sistema. Valida que todos los campos sean llenados.
3.		Termina el caso de uso.
Flujos alternos		
1. Los datos son incorrectos		
	Actor	Sistema
1.	Introduce cadenas no compatibles con el tipo de datos o demasiado largas.	Muestra mensaje de error indicando que la cadena de caracteres no es compatible o que ya existe un contenido con esa propiedad.
2.		Termina el caso de uso.
2. Los campos están vacíos.		
	Actor	Sistema
1.	Ha dejado campos vacíos a la hora de adicionar el contenido.	Mostrará un mensaje de error que le señala al usuario que ese campo no puede estar vacío y continúa en la página inicial.
2.		Termina el caso de uso.
Relaciones	CU Incluidos	

	CU Extendidos	
Requisitos funcionales	no	
Asuntos pendientes		

Tabla 9. Especificación de requisitos CU Adicionar Contenidos.

CU_3. Editar Contenidos

Objetivo	Editar contenido existente en el sistema.	
Actores	Administrador: (Inicia) Edita el contenido de cualquier espacio del sistema. Editor:(Inicia) Edita contenido del sistema. Gestor:(Inicia) Edita contenido del sistema.	
Resumen	El CU permite editar un contenido del sistema.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Usuario del sistema autenticado.	
Postcondiciones	Contenido editado en el sistema.	
Flujo de eventos		
Flujo básico Editar contenidos.		
	Actor	Sistema
1.	Para editar un contenido debe estar autenticado en el sistema. Una vez que acceda a su espacio de trabajo, según los permisos asignados, puede editar uno o varios contenidos. Selecciona un tipo de contenido.	Muestra los detalles del contenido seleccionado por el usuario con la opción de edición deshabilitada.
2.	Selecciona la opción de editar ese contenido.	Muestra formulario con los campos editables activos para que el usuario efectúe las modificaciones o actualizaciones.
3.	Introduce los cambios o actualizaciones que desea realizar al contenido.	Valida que los campos estén completos.
4.		Guarda los cambios en la aplicación.
5.		Termina el caso de uso.
Flujos alternos		
1. Los datos son incorrectos		
	Actor	Sistema
1.	Introduce cadenas no compatibles con el tipo de datos o demasiado largas.	Muestra mensaje de error indicando que la cadena de caracteres no es compatible o que ya existe un archivo con esa propiedad.

2.		Termina el caso de uso.
2. Los campos están vacíos.		
	Actor	Sistema
1.	Ha dejado campos vacíos a la hora de editar el contenido.	Mostrará un mensaje de error que le señala al usuario que ese campo no puede estar vacío y continúa en la página de edición.
2.		Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos funcionales	no	
Asuntos pendientes		

Tabla 10. Especificación de requisitos CU Editar Contenidos.

CU_4. Eliminar Contenidos

Objetivo	Eliminar contenidos del sistema.	
Actores	Administrador: (Inicia) Elimina un contenido del sistema. Colaborador: (Inicia) Elimina un contenido del sistema. Editor: (Inicia) Elimina un contenido del sistema. Gestor: (Inicia) Elimina un contenido del sistema.	
Resumen	El CU permite eliminar un contenido del sistema.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Usuario del sistema autenticado.	
Postcondiciones	Contenido eliminado del sistema.	
Flujo de eventos		
Flujo básico Eliminar contenidos.		
	Actor	Sistema
1.	Para la eliminación de contenido debe estar autenticado en el sistema. Una vez que acceda a su espacio de trabajo, según los permisos asignados, puede eliminar una o varios contenidos.	Mostrará mensaje para confirmar si realmente desea eliminar el contenido.
2.	Confirma la eliminación del contenido o la cancela.	Elimina el contenido de la base de datos o restablece las condiciones anteriores cerrando el formulario de confirmación.
3.		Termina el caso de uso.
Relaciones	CU Incluidos	

	CU Extendidos	
Requisitos funcionales	no	
Asuntos pendientes		

Tabla 11. Especificación de requisitos CU Eliminar Contenidos.

Anexo 3. Estándares de codificación

Formato de archivos PHP

General: Para archivos que contengan únicamente código PHP, la etiqueta de cierre ("?>") no está permitida. No es requerida por PHP, y omitirla evita la inyección de espacios en blanco en la respuesta.

Indentación: La indentación está compuesta por 4 espacios. Las tabulaciones no están permitidas.

Tamaño máximo de línea: El tamaño máximo de cualquier línea de código PHP es de 160 caracteres.

Final de línea: El Final de Línea sigue la convención de archivos de texto Unix. Las líneas deben acabar con un carácter *linefeed* (LF).

Convenciones de Nombres

Clases: Los nombres de clases pueden contener sólo caracteres alfanuméricos. Los números están permitidos en los nombres de clase, pero desaconsejados en la mayoría de casos. Las barras bajas (_) están permitidas solo como separador de ruta (el archivo " Zend/Db/Table.php " debe apuntar al nombre de clase " Zend_Db_Table ").

Si el nombre de una clase está compuesto por más de una palabra, la primera letra de cada palabra debe aparecer en mayúsculas. Poner en mayúsculas las letras siguientes no está permitido, ej: "Zend_PDF" no está permitido, mientras que " Zend_Pdf " es admisible.

Funciones y Métodos: Los nombres de funciones pueden contener únicamente caracteres alfanuméricos. Los guiones bajos (_) no están permitidos. Los números están permitidos en los nombres de función pero no se aconseja en la mayoría de los casos.

Los nombres de funciones deben empezar siempre con una letra minúscula. Cuando un nombre de función consiste en más de una palabra, la primera letra de cada nueva palabra debe estar en mayúsculas. Esto es llamado comúnmente como formato "camelCase".

Estos son ejemplos de nombres de funciones admisibles:

```
filterInput()
```

```
getElementById()
```

```
widgetFactory()
```

Para la programación orientada a objetos, los métodos de acceso para las instancias o variables estáticas

deben ir antepuestos con un "get" o un "set".

Estilo de código

Demarcación de código PHP: El código PHP debe estar delimitado siempre por la forma completa de las etiquetas PHP estándar:

```
<?php
?>
```

Las etiquetas cortas (*short tags*) no se permiten nunca. Para archivos que contengan únicamente código PHP, la etiqueta de cierre debe omitirse siempre.

Clases

Declaración de clases

Las Clases deben ser nombradas de acuerdo a las convenciones de nombres. La llave "{" deberá escribirse siempre en la línea debajo del nombre de la clase. Todo el código contenido en una clase debe ser separado con cuatro espacios.

Únicamente una clase está permitida por archivo PHP. En archivos de ese tipo, dos líneas en blanco deben separar la clase de cualquier código PHP adicional en el archivo de clase.

A continuación se muestra un ejemplo de una declaración de clase que es permitida:

```
/**
 * Bloque de Documentación aquí
 */
class SampleClass
{
    // el contenido de la clase
    // debe separarse con cuatro espacios
}
```

Las clases que extiendan otras clases o interfaces deberían declarar sus dependencias en la misma línea siempre que sea posible.

```
class SampleClass extends FooAbstract implements BarInterface
{
}
```

Funciones y Métodos

Declaración de Funciones y Métodos: Las Funciones deben ser nombradas de acuerdo a las convenciones de nombrado establecido. Los métodos dentro de clases deben declarar siempre su visibilidad usando un modificador *private*, *protected*, o *public*.

Como en las clases, la llave "{" debe ser escrita en la línea siguiente al nombre de la función ("*one true brace*" form). No está permitido un espacio entre el nombre de la función y el paréntesis de apertura para los argumentos.

Las funciones de alcance global no están permitidas.

Lo siguiente es un ejemplo de una declaración admisible de una función en una clase:

```
/**
 * Bloque de Documentación aquí
 */
class Foo
{
    /**
     * Bloque de Documentación aquí
     */
    public function bar()
    {
        // el contenido de la función
        // debe separarse con cuatro espacios
    }
}
```

Anexo 4. Pantallas principales de la aplicación

Búsqueda avanzada [icon] [icon] [icon] [icon] [input] [input] usuario c0ntr@s3ña [icon]

TeVeó Plus Español English [icon] [icon] [icon]

Noticia del día 5 5 0

[Video Player: A group of dancers performing on stage]

[ver más](#)

Te recomendamos

- Media heading tipo descripción
- Media heading tipo descripción
- Media heading tipo descripción

Más populares

- Media heading tipo descripción 150
- Media heading tipo descripción 100
- Media heading tipo descripción 50

Recientes

- Media heading tipo descripción
- Media heading tipo descripción
- Media heading tipo descripción

UCI **Copyleft** Optimizado para Firefox 19 Resolución 1024x768

Figura 18. Vista inicial de la aplicación.

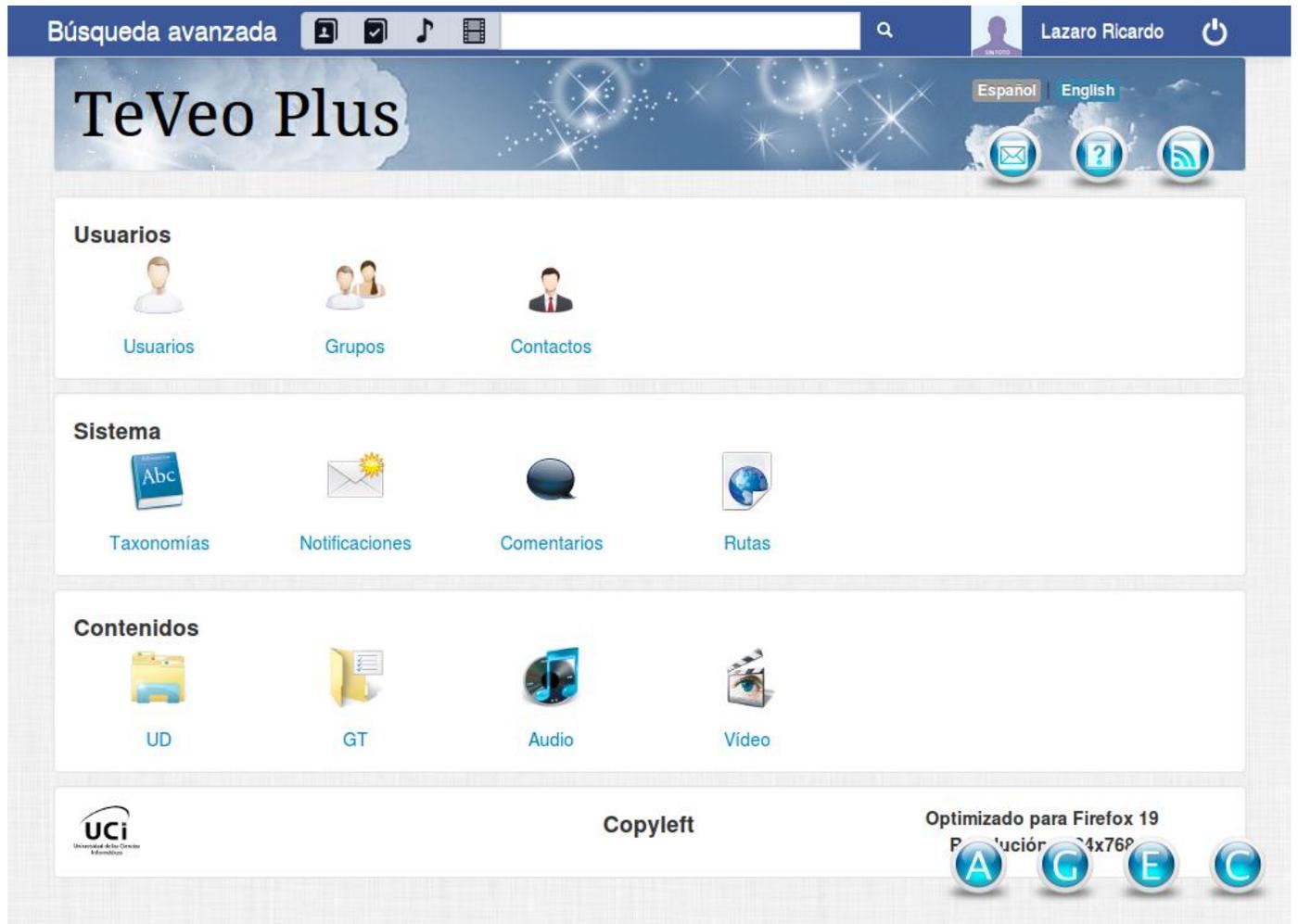


Figura 19. Vista de administración.

Anexo 5. Diseño de casos de prueba basados en casos de uso

Adicionar contenido

Condiciones de ejecución:

- El usuario debe estar autenticado en el sistema con el rol de colaborador o gestor.

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo central
Adicionar contenido.	Adición de un contenido con nombre válido en el sistema.	V	Adición del contenido al espacio personal del usuario autenticado.	<ul style="list-style-type: none"> - Presionar la opción adicionar contenido. - Introducir los datos asociados de forma correcta. - Oprimir el botón "Crear".
Mostrar mensaje de error de nombre repetido.	Nombre del contenido existente en el sistema.	I	Mensaje de error que especifica "Ya existe un contenido con ese nombre".	<ul style="list-style-type: none"> - Presionar la opción adicionar contenido. - Introducir entre los datos asociados un nombre que haya sido adicionado anteriormente. - Oprimir el botón "Crear".
Mostrar mensaje de campo con caracteres no válidos.	Campos con caracteres no válidos.	I	Mensaje de error que especifica "Formato no válido."	<ul style="list-style-type: none"> - Presionar la opción adicionar contenido. - Introducir entre los datos asociados un nombre con caracteres extraños. - Oprimir el botón "Crear".
Mostrar mensaje de nombre demasiado pequeño.	Nombre del contenido con menos de 3 caracteres.	I	Mensaje de error que especifica "El nombre debe contener entre 3 y 30 caracteres."	<ul style="list-style-type: none"> - Presionar la opción adicionar contenido. - Introducir entre los datos asociados un nombre con 1 o 2 caracteres. - Oprimir el botón "Crear".
Mostrar mensaje de nombre demasiado extenso.	Nombre del contenido con más de 30 caracteres.	I	Mensaje de error que especifica "El nombre debe contener entre 3 y 30 caracteres."	<ul style="list-style-type: none"> - Presionar la opción adicionar contenido. - Introducir entre los datos asociados un nombre con más de 30 caracteres.

				- Oprimir el botón "Crear".
Mostrar mensaje de campo vacío.	Campo nombre del contenido, vacío.	NA	Mensaje de error que especifica "Campo requerido."	- Presionar la opción adicionar contenido. - Introducir el nombre del contenido - Oprimir el botón "Crear".

Tabla 12. Diseño de caso de prueba del caso de uso Adicionar contenido.

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto.	No	Máximo 30 caracteres. Mínimo 3 caracteres. No admite caracteres extraños (@, #, ~, \$, %, &, /, ¬).

Tabla 13. Variables empleadas en el diseño de caso de prueba Adicionar contenido.

Eliminar contenido

Condiciones de ejecución:

- El usuario debe estar autenticado en el sistema con el rol de colaborador o gestor.

Escenario	Descripción	Respuesta del sistema	Flujo central
Eliminar contenido.	Se seleccionan los contenidos y se oprime el botón "eliminar".	- Si el usuario acepta muestra un mensaje de confirmación "El contenido se ha eliminado correctamente", luego se eliminan los contenidos seleccionados y se recarga el bloque de contenidos del espacio personal del usuario. - En caso contrario, muestra un mensaje informativo "No se ha podido eliminar el contenido", luego vuelve a la página inicial del usuario.	- Realizar la búsqueda del contenido que se desea eliminar. - Seleccionar los contenidos a eliminar. - Presionar el ícono "Eliminar".

Tabla 14. Diseño de caso de prueba del caso de uso Eliminar contenido.