



Módulo para la visualización de la información de las pistas de auditorías en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.

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

Autores:

Reynaldo Rodríguez López
Hamdy Abdelhay Mohamed Martínez

Tutores:

Ing. Yamila Díaz Suárez
Ing. Arianne Valdés Álvarez

Consultor:

Lic. Alejandro J. Rodríguez Aguila

Declaramos que somos los únicos autores del trabajo titulado: Módulo para la visualización de la información de las pistas de auditorías en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.

Se autoriza a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de ____ del año 2016.

Reynaldo Rodríguez López

Hamdy Abdelhay Mohamed Martínez

Ing. Yamila Díaz Suárez

Ing. Arianne Valdés Álvarez

Lic. Alejandro J. Rodríguez Aguila

En nombre de los dos autores, agradecer a la Universidad de las Ciencias Informáticas por todos los conocimientos dados y por la formación como profesionales.

A los tutores por guiarnos en el desarrollo del trabajo de diploma, al arquitecto del proyecto eXcriba Oscar Daniel por su gran ayuda en la implementación de la investigación y a todos los buenos profesores que han pasados por nuestras aulas.

Reynaldo R.L

A mis padres por estar siempre junto a mí.

A mi papá por ser tan fuerte y cariñoso a la vez, por enseñarme a ser el hombre que soy y por saber formar una familia tan unida.

A mi mamá por ser tan dulce y por hacerme entender la importancia de seguir adelante a pesar de los obstáculos.

A mi hermana por apoyarme y aconsejarme en todas las decisiones, a Divisito por alegrar los días que estoy en casa.

A mi novia por aguantar mi lado más dulce y más amargo

A mi compañero de tesis.

A todos mis amigos y compañeros, en especial a mi gran amigo y hermano que ha estado conmigo desde los 6 años, a sus preciosos niños y a su gran esposa.

En fin, a todo por apoyarme en todos estos años de la carrera. Gracias.

Hamdy A.

A mi mamá por ser esa persona que siempre ha luchado por mí desde que tengo memoria, por no ser solamente madre, sino también padre en los momentos que lo necesitaba, por todos los sacrificios que tuvo que hacer para que mi hermana y yo seamos hoy las personas que somos, por apoyarme en todas las decisiones que tomé, aunque no estuviera de acuerdo con alguna de ellas. Eres lo más grande que tengo en la vida y no hay palabras suficientes para dedicarte.

A mi hermana, a pesar que somos primos y siempre nos estamos fajando eres una de las personas más importante para mí, he pasado toda la vida contigo y por eso te quiero como una hermana.

A mi tío Noel por apoyarme en estos cinco años de universidad en lo que necesitara y siempre confiar en mí. Por tratarme y hablarme como a un hijo cuando era necesario. Fuiste un pilar muy importante en mi superación personal y profesional. Gracias.

A todos mis familiares que están aquí en estos momentos.

A mis hermanos de la secundaria y el pre, Leosney, Salien, Dalia, Laimber y Ángel Luis.

A todos mis compañeros de aula y edificio, desde que empecé en primer año en el grupo 2102 y el edificio 60 hasta ahora en el grupo 2501 y el edificio 15. Son demasiados para mencionarlos todos.

A Yazmin por ser una excelente amiga, solo compartimos poco tiempo, pero te convertiste en alguien especial, aunque no lo creas me enseñaste mucho acerca de la vida.

A Rey, mi compañero de tesis, por querer compartir esta experiencia conmigo y a su novia Yanet por apoyarnos.

A todos los que en algún momento me preguntaron ¿Cómo te va la tesis?

Reynaldo R.L

A mis padres y a todos los que de una forma u otra me ayudaron en los años de la carrera.

Hamdy A.

A mi mamá, por dedicarse por completo a mí todos estos años.

A mi hermana y mis sobrinos, los quiero mucho.

A mi abuela y a tío Enrique, que no están presentes físicamente, pero los llevo en el corazón.

A Melo, a pesar que no era parte de mi familia la quería como si lo fuera.

A todos mis familiares y amigos.

Al padre de mi compañero de tesis.

En la Universidad de las Ciencias Informáticas, centro que se caracteriza por su preparación docente-productiva, se encuentran varios centros de producción, entre ellos el Centro de Informatización de la Gestión Documental. En este centro se desarrolla el Gestor de Documentos Administrativos Xabal eXcriba 3.1. El *software* en la actualidad no cuenta con un mecanismo para visualizar las pistas de auditorías a nivel de usuario generadas por el Gestor de Contenidos Empresarial Alfresco 4.2f, imposibilitando mantener un seguimiento de las acciones que son realizadas por los usuarios sobre los documentos, sus registros de sesión y los flujos de trabajo, dificultando el análisis de los datos para la toma de decisiones.

La investigación que se llevó a cabo propone el desarrollo de un módulo que permite visualizar a nivel de usuario las pistas de auditoría generadas por Alfresco. En el desarrollo del módulo se utiliza la API¹ de JavaScript de Alfresco para acceder al repositorio² donde se encuentran las pistas de auditorías, el motor de plantillas *FreeMarker*³ y la elaboración de web scripts⁴. Con este módulo se espera que el Gestor de Documentos Administrativos Xabal eXcriba 3.1 proporcione un mejor análisis de la información del sistema para la toma de decisiones en las organizaciones, así como mantener un seguimiento sobre las actividades realizadas por un usuario en el sistema.

Palabras clave: Gestor de Documentos Administrativos Xabal eXcriba, Gestor de contenidos Empresariales Alfresco, sistema de gestión documental, pistas de auditorías.

¹ Interfaz de Programación de Aplicaciones, por sus siglas en inglés (API). Es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción.

² Depósito de archivos digitales en el que se almacena toda la información aportada por los usuarios.

³ Motor de plantillas basado en Java que se centra en la arquitectura de *software* Modelo Vista Controlador. Es utilizado para generar la salida de los datos que proporcionan los servicios web.

⁴ Los web scripts en Alfresco proporcionan una forma única de interactuar mediante programación con el servidor de aplicaciones de contenido de Alfresco.

Capítulo 1: Fundamentación teórica sobre las pistas de auditorías en los sistemas de gestión documental	16
1.1 Conceptos y aspectos relacionados con la investigación.....	16
Gestión documental.....	16
Sistema de gestión documental.....	17
Auditorías	18
Las pistas de auditorías en los SGD.....	19
1.2 Normas y recomendaciones sobre sistemas de gestión documental y las pistas de auditorías	20
1.3 Pistas de auditorías en Alfresco.....	21
1.4 Herramientas para la visualización de las pistas de auditorías en Alfresco y otros sistemas	22
1.5 Gestor de Documentos Administrativos Xabal eXcriba.....	28
Conclusiones parciales.....	29
Capítulo 2: Características de la solución.....	30
2.1 Metodología de desarrollo de <i>software</i>.....	30
2.2 Lenguaje de modelado UML 2.1.....	33
2.3 Herramienta de modelado Visual Paradigm.....	33
2.4 Modelado de dominio	33
2.4.1 Conceptos fundamentales del dominio	34
2.5 Especificación de los requisitos del <i>software</i>.....	35
2.5.1 Requisitos funcionales	35
2.5.2 Requisitos no funcionales	37
2.6 Actores del sistema	37
2.7 Historias de Usuario	37
2.8 Plan de iteraciones	41
2.9 Arquitectura de la solución.....	43
2.9.1 Arquitectura en capas	43

2.10 Patrón de diseño.....	44
2.10.1 Patrones GRASP	44
Conclusiones Parciales.....	45
Capítulo 3: Implementación y validación de la solución	46
3.1 Descripción de las herramientas, técnicas y lenguajes utilizados.....	46
3.1.1 JavaScript versión 1.6.....	46
3.1.2 API JavaScript para Alfresco versión 4.0	46
3.1.3 AJAX.....	46
3.1.4 HTML versión 5.....	47
3.1.5 CSS versión 3	47
3.1.6 XML	48
3.1.7 <i>API Template de Alfresco</i>	48
3.1.8 Editor de código Visual Studio Code.....	48
3.2 Bibliotecas utilizadas.....	49
3.2.1 JQuery (versión 2.2.3)	49
3.2.2 JQuery UI (versión 1.8.11).....	49
3.2.3 Biblioteca de gráficas amCharts versión 3.12.2.....	49
3.3 Propuesta del módulo	49
3.4 Estrategia de prueba	50
Conclusiones parciales	55
Conclusiones generales	56
Referencias Bibliográficas	58
Bibliografía	61

Figura 1: Visualización de las pistas de auditoría en Audit Surf.....	24
Figura 2 Visualización de las pistas de auditoría en Audit Dashlet	25
Figura 3 Visualización de las pistas de auditoría en Audit Share.....	26
Figura 4: Visualización de las pistas de auditorías en AAAR.....	27
Figura 5: Visualización de las pistas de auditorías en Formtek Auditing Extension.	28
Figura 6: Fases y disciplinas de AUP.	30
Figura 7: Modelado del Dominio.....	34

Tabla 1: Actor del sistema.....	37
Tabla 2: HU #01 “Mostrar todas las acciones de un usuario sobre los documentos”.....	39
Tabla 3: HU #11 “Mostrar todos los registros de sesión de un usuario en el sistema”.....	40
Tabla 4: HU #16 “Mostrar los flujos de trabajos por el usuario iniciador en estado activos”.....	41
Tabla 5 Descripción de las variables.....	52
Tabla 6: Descripción del caso de prueba # 01 de la HU #01.....	53
Tabla 7 Descripción del caso de prueba # 02 de la HU #11.....	53
Tabla 8 Descripción del caso de prueba # 03 de la HU #16.....	54

El almacenamiento no controlado de cualquier tipo de documento, genera la necesidad de implantar soluciones capaces de procesar toda esta documentación, organizarla y, sobre todo, poder extraer de ella un valor añadido. La gestión documental surge hacia la década de los años cuarenta del siglo XX, debido a la preocupación de los gobiernos por el aumento indistinto de documentos a los que no se brindaba tratamiento alguno (1). La gestión documental se ha convertido actualmente en una necesidad para la mayoría de las organizaciones, pues estas necesitan mantener ordenada y accesible toda la información que resulte de valor para ellas. Con el objetivo de automatizar estas acciones y sobre todo facilitar la consulta y acceso desde cualquier lugar, se crearon aplicaciones informáticas conocidas como Sistemas de Gestión Documental (SGD).

Para supervisar el funcionamiento correcto de los SGD, existen normas e informes técnicos por los cuales se deben regir, como son las normas ISO⁵ 15489 y 30 300, además de las especificaciones MoReq⁶. Estos sistemas deben mantener un control estricto sobre todas las acciones que se realizan en ellos, prestando especial atención a las que se ejecutan sobre los documentos, quién los crea, quién trabaja con ellos y quién los modifica, permitiendo registrar y evaluar los flujos de información de estos documentos. Para dar cumplimiento a esto, se propone la utilización de las *audit trails*, que en términos de español serían pistas de auditorías. Estas pistas de auditorías se consideran un registro que dejan las acciones realizadas por los usuarios y las iniciadas de forma automática por el sistema (2).

En Cuba el tema de la gestión documental se ha tratado de manera aislada por unos pocos autores, entre los que se pueden mencionar Mayra Mena⁷ y Gloria Ponjuán⁸. Estos autores plantean que debido a las nuevas formas de organización y funcionamiento de las empresas cubanas, y el proceso de introducción de las TICs⁹ se prevé un incremento en la documentación de forma electrónica. Sin embargo, llegan a la conclusión que en las organizaciones cubanas no existe una correcta identificación del papel de la gestión documental para asegurar la información de las organizaciones. Los estudios realizados por estos autores dieron como resultado que no existen las herramientas e instrumentos que permitan el manejo de los documentos electrónicos y que sean capaces de garantizar la veracidad, autenticidad y confiabilidad de los mismos, además de la falta de homogenización en la implementación de estos SGD entre las diferentes empresas del país. Estas

⁵ Organización Internacional de Normalización ISO.

⁶ Modelo de Requisitos para la Gestión de Documentos Electrónicos de Archivo.

⁷ Doctora en Ciencias de la Información. Departamento de Ciencias de la Información. Facultad de Comunicación. Universidad de La Habana, Cuba.

⁸ Doctora en Bibliotecología y Ciencia de la Información por la Facultad de Comunicación de la Universidad de La Habana (2000), Licenciada en Información Científico Técnica por la Universidad de La Habana (1977).

⁹ Tecnologías de la Información y las Comunicaciones.

deficiencias mencionadas provocan la pérdida de grandes cantidades de documentación e información, imposibilitando entender quién sabe y qué sabe la empresa. Además, provoca dificultades para socializar y compartir conocimiento, obstaculizando la colaboración interpersonal e interinstitucional (3).

La Universidad de las Ciencias Informáticas (UCI), entidad que se caracteriza por su preparación educativa y productiva, brinda servicios informáticos y desarrollo de aplicaciones a diversos sectores de la economía y servicios, tanto dentro como fuera del país. Para lograr esto se apoya en los centros de desarrollo existentes en la universidad, vinculando estudio-trabajo mediante las actividades realizadas en estos centros de desarrollo. Entre ellos se encuentra el Centro de Informatización de la Gestión Documental (CIGED), donde se desarrolla el Gestor de Documentos Administrativos (GDA) eXcriba, con el fin de automatizar los procesos documentales y archivísticos que se ejecutan dentro de cualquier entidad. Actualmente el *software* se encuentra en su versión 3.1, comercializado bajo la marca Xabal. El sistema permite realizar un conjunto de acciones sobre los documentos, tales como crearlos y clasificarlos, mantener un control de versiones sobre ellos, definir tipologías documentales, automatizar los flujos documentales, compartir los documentos y almacenarlos en diferentes formatos electrónicos, entre otras.

Xabal eXcriba 3.1 cuenta con un conjunto de *dashlet*¹⁰ que permiten mostrar información relacionada a las actividades realizadas por los usuarios y las tareas que tienen pendientes, también, muestra quién ha modificado un documento sobre el que trabaja, las áreas a las que se tiene acceso y los miembros del área, entre otras. Esta información proporciona una idea preliminar y sintetizada sobre lo que ocurre en la aplicación, sin embargo, se desea mantener un mayor seguimiento de las acciones que se realizan en el sistema, planteándose la incorporación de nuevas funcionalidades que le permitan al administrador conocer los rastros que dejan los usuarios en el sistema, así como el seguimiento de los documentos desde que son creados hasta su expurgo. En la actualidad, eXcriba no cuenta con un mecanismo capaz de visualizar a nivel de usuario la información referente a las pistas de auditorías, imposibilitando mantener un seguimiento de las acciones realizadas por los usuarios sobre los documentos, los registros de sesión por cada uno y los flujos de trabajos activos y completados en el sistema, dificultando el análisis de los datos necesarios para la toma de decisiones.

¹⁰ Un *dashlet* es un módulo o miniaplicación que permite obtener información sobre los elementos de interés para los usuarios del sistema.

Luego de analizar la situación existente, se identifica como **problema de investigación**: ¿cómo contribuir al seguimiento y visualización de las pistas de auditorías de los documentos, actividades de usuarios y flujos de trabajo en el Gestor de Documentos Administrativo Xabal eXcriba 3.1?

Del problema planteado se propone como **objeto de estudio**: el tratamiento y visualización de las pistas de auditorías en los sistemas de gestión documental, con el **objetivo general**: desarrollar un módulo para la visualización de la información de las pistas de auditorías en el Gestor de Documentos Administrativo Xabal eXcriba 3.1, delimitando como **campo de acción**: el tratamiento y visualización de las pistas de auditorías en el Gestor de Documentos Administrativo Xabal eXcriba 3.1.

Para dar cumplimiento al objetivo propuesto se derivan los siguientes **objetivos específicos**:

1. Analizar las bases teóricas-conceptuales sobre las pistas de auditorías en los sistemas de gestión documental para identificar características generales y mecanismos de visualización utilizados por otras herramientas.
2. Diseñar un módulo utilizando la metodología de desarrollo de *software* para definir los principales componentes de su arquitectura.
3. Implementar un módulo utilizando el diseño, las herramientas y lenguajes definidos en la investigación brindando una solución al problema planteado.
4. Validar la solución propuesta mediante pruebas de *software* de caja negra para comprobar el correcto funcionamiento de la solución obtenida.

Métodos de investigación:

Para dar cumplimiento a los objetivos planteados se aplican varios métodos de investigación en la búsqueda y procesamiento de la información, como son:

Métodos teóricos:

- **Histórico-Lógico**: permite la comprensión de investigaciones anteriores sobre las pistas de auditorías en los sistemas de gestión documental.
- **Analítico-Sintético**: se emplea para analizar los distintos materiales bibliográficos relacionados con la investigación para sintetizar las características fundamentales del desarrollo de la propuesta de solución.
- **Modelación**: se utiliza para analizar la solución que se propone mediante el diagrama de dominio en la etapa de diseño para un mejor entendimiento de la propuesta de solución.

Métodos empíricos:

- **Análisis documental:** se utiliza en las consultas de la literatura especializada de los sistemas de gestión documental y las pistas de auditorías en los sistemas de gestión documental.
- **Observación:** se emplea para conocer las características y funcionamiento de los sistemas homólogos, permitiendo obtener la información referente a la visualización de las pistas de auditorías en los sistemas de gestión documental.

Estructura capitular:

El presente trabajo de investigación está estructurado en 3 capítulos, de los cuales se brinda una breve descripción a continuación:

- **Capítulo 1- Fundamentación teórica sobre las pistas de auditorías en los sistemas de gestión documental:** en este capítulo se introducen los conceptos más importantes relacionados con las pistas de auditorías en los sistemas de gestión documental. Se realiza un estudio de algunas herramientas para la visualización de las pistas de auditorías y posibles soluciones existentes, así como de las normas e informes técnicos por las que rigen su funcionamiento.
- **Capítulo 2- Características de la solución:** se caracteriza la metodología de desarrollo a utilizar, así como el lenguaje y la herramienta de modelado. Se identifican los requisitos funcionales y no funcionales del sistema, además, se presentan las descripciones de las historias de usuario y se describe la arquitectura y los patrones de diseño.
- **Capítulo 3- Implementación y validación de la solución:** se caracterizan las herramientas, técnicas y lenguajes utilizados. También se describe la solución que se implementa y se detalla la estrategia de prueba a seguir para la validación de la solución.

Fundamentación teórica sobre las pistas de auditorías en los sistemas de gestión documental

En este capítulo se presentan los elementos teóricos conceptuales que sustentan al problema de investigación. Se realiza un estudio y análisis de los conceptos asociados a la gestión documental y las pistas de auditorías en los sistemas de gestión documental, así como de las principales normas e informes técnicos que las rigen. Se realiza el estudio de varias herramientas relacionadas con la visualización de las pistas de auditorías en los sistemas de gestión documental.

1.1 Conceptos y aspectos relacionados con la investigación

Para lograr una mejor comprensión del contexto en el que se enmarca el problema, es necesario definir y describir brevemente algunos conceptos asociados, entre los que se encuentran:

Gestión documental

La gestión documental es el área de gestión responsable de un control eficaz y sistemático de la creación, la recepción, el mantenimiento, el uso y la disposición de documentos, incluidos los procesos para incorporar y mantener en forma de documentos la información y prueba de las actividades y operaciones de la organización (4).

Una adecuada gestión documental constituye una base sólida para que una organización logre eficiencia en la toma de decisiones, transparencia, rendición de cuentas, control interno, y la protección de los derechos ciudadanos. Además, garantiza la salvaguarda de la memoria institucional y la protección del patrimonio documental (5).

El Diccionario de Terminología Archivística editado por el Consejo Internacional de Archivos la define como “*Un aspecto de la Administración General relacionado con la búsqueda de la economía y eficacia en la producción, mantenimiento, uso y destino final de los documentos*” (6).

Así mismo la gestión documental es “*el proceso que administra el ciclo vital de los documentos, es decir, desde su producción hasta su eliminación o conservación permanente. Regula la trazabilidad de los documentos mediante normas, técnicas y buenas prácticas archivísticas. Abarca la producción, organización (clasificación, ordenación y descripción), conservación, automatización, uso, selección y eliminación documental*” (7).

Mayra Mena en su libro “*Gestión documental y organización de archivo*” expresa que la gestión de documentos “*engloba un conjunto de operaciones comprometidas en la búsqueda de la economía y la eficacia en la producción, mantenimiento, uso y destino final de los documentos a lo largo de todo su ciclo de vida*” (1).

Considerando las definiciones antes abordadas se considera que la gestión documental es un conjunto de normas, tecnologías y técnicas que le permiten a las organizaciones administrar el flujo de documentos que se generan en ellas, llevando los mismos a lo largo de su ciclo de vida, con el objetivo de facilitar su utilización y conservación, ya sea de forma manual o basándose en herramientas informáticas.

Hoy en día, las empresas u organizaciones trabajan con gran cantidad de documentación. Con el fin de facilitar la gestión de estos, aparecen los sistemas de gestión documental permitiendo de manera eficiente el fácil almacenamiento y búsqueda rápida de la información.

Sistema de gestión documental

Atendiendo al objetivo de la gestión documental de compartir la información para hacer seguros los documentos, accesibles, recuperables e intercambiables, emerge la solución para esta situación, los Sistemas de Gestión Documental.

Un sistema de gestión documental *“es un sistema que permite la automatización, la creación, el mantenimiento y la consulta de fuentes de información constituidas por documentos y, por lo tanto, sirve para explotar el conocimiento que contienen los documentos con el fin de ponerlo al alcance de los usuarios del sistema”* (8).

La norma ISO 15489 define los sistemas de gestión documental como *“un sistema de información que incorpora, gestiona y facilita el acceso a los documentos a lo largo del tiempo”* (4).

Mientras que la norma ISO 30300 lo define *“como un sistema de gestión para dirigir y controlar una organización en lo relativo a los documentos”* (9).

El principal propósito de los SGD es proveer un repositorio donde los documentos pueden ser creados, clasificados, manejados y almacenados para facilitar el acceso de usuarios a los mismos. Representan la integración de las tecnologías de la información con el tratamiento de documentos que incluye almacenamiento, manejo, recuperación y acceso a los documentos. Los SGD permiten que los documentos sean procesados electrónicamente y de forma más eficiente que el manejo de los mismos en formato duro y que puedan tratarse otros tipos de archivos como audio, imágenes y videos (10).

A partir de los conceptos anteriormente planteados, se considera que los sistemas de gestión documental son herramientas que le permiten a una entidad mantener un seguimiento sobre el ciclo de vida de los documentos que se consideran de importancia para ellas. Su aplicación permite la obtención de una mejor organización y utilización de los documentos electrónicos acumulados por la entidad, producto del desarrollo de sus funciones.

Auditorías

Inicialmente, la auditoría se limitaba a las verificaciones de los registros contables, dedicándose a observar si los mismos eran exactos. Con el tiempo, el campo de acción de la auditoría ha continuado extendiéndose a otros campos profesionales, como la administración, ingeniería, medicina, etc. hasta alcanzar casi todas las disciplinas del trabajo.

Según la *American Accounting Association (AAS)* “auditoría es el proceso sistemático para obtener y evaluar de manera objetiva las evidencias relacionadas con informes sobre actividades económicas y otros acontecimientos relacionados” (10).

Alberto Salinas La Rosa la define como “la revisión independiente de una actividad o un conjunto de actividades, funciones específicas, resultados u operaciones de una entidad administrativa, realizada por un profesional de la auditoría, con el propósito de evaluar su correcta realización y, con base en ese análisis, poder emitir una opinión autorizada sobre la racionalidad de sus resultados y el cumplimiento de sus operaciones” (11).

Según Ricardo Vilches Troncoso de la Universidad Católica Cardenal Raúl Silva, define auditoría como “una sistemática evaluación de las diversas operaciones y controles de una organización, para determinar si se siguen políticas y procedimientos aceptables, si se siguen normas establecidas, si se utilizan los recursos eficientemente y si se han alcanzado los objetivos de la organización” (12).

Dadas las definiciones antes abordadas se puede concretar que las auditorías se realizan con el objetivo de verificar y comprobar la veracidad de los hechos según lo que está establecido en las normas y regulaciones de cualquier sector.

El término auditoría tiene diversos tipos de enfoques, dependiendo del criterio y finalidad de la información. Entre ellas se pueden mencionar auditorías internas, externas, auditorías de gestión, organizativas, operativas, financieras, informáticas, entre otras. Con el objetivo de analizar las auditorías para esta investigación se profundiza en las auditorías informáticas o de sistemas.

“La auditoría de sistema es el proceso de recoger, agrupar y evaluar evidencias para determinar si un sistema de información salvaguarda el activo empresarial, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización, utiliza eficientemente los recursos y cumple con las leyes y regulaciones establecidas” (11).

Según Charles D. Ellis¹¹ auditoría de información es la revisión de los sistemas de gestión de información existentes, con el objetivo de identificar problemas y encontrar soluciones a dichos problemas (13).

También Buchanan, S. y Gibb, F. la define como el análisis de los procesos de información que tienen lugar entre los integrantes de una organización y los medios o canales utilizados en los procesos comunicativos (14).

Coincidiendo con las valoraciones antes planteadas se puede decir que una auditoría de sistema es una herramienta que permite mantener un seguimiento sobre las actividades de una organización.

Los sistemas de gestión documental deben ser capaces de seguir y gestionar los documentos que se encuentran en ellos, de manera que se cumplan requisitos de seguridad. Para esto las pistas de auditorías son un elemento clave, puesto que registran de forma exhaustiva todas las acciones sobre cualquier documento (2).

Las pistas de auditorías en los SGD

Las pistas de auditorías en inglés *audit trails* según Ricardo J. Castellano¹² son elementos que permiten certificar la existencia de una operación, la validez de sus cifras, la identidad de los sujetos involucrados, entre otros, dejando evidenciada una transacción. (15).

MoReq la define como “*la información sobre las transacciones u otras actividades que hayan influido o modificado entidades*”¹³ y que aporta detalles suficientes para permitir la reconstrucción de la actividad anterior” (2).

Para Sergio Espinosa Guido¹⁴ las pistas de auditorías no son controles por sí mismos, son rastros que se requieren para el uso analítico y administrativo, son más que todo un procedimiento que puede estar constituido por uno o varios documentos, informes o simplemente controles o conjunto de controles. (16).

¹¹ Consultor de inversiones americano fundador del Greenwich Associates, firma de consultoría a las instituciones financieras.

¹² Profesor Auditoría de Sistemas Computarizados de la Facultad de Ciencias Económicas-UNC.

¹³ MoReq define entidades a los usuarios o grupos de usuarios en un Sistema de Gestión de Documentos Electrónicos.

¹⁴ Doctor en Ciencias de la Educación; Máster en Auditoría Informática; Licenciado en Administración de Negocios. Profesor en la Universidad de Costa Rica; en el Instituto Tecnológico de Costa Rica y en la Maestría de Auditoría Informática en la UCR. Autor de varios artículos relacionados con sus especialidades y de libro titulado: “Auditoría de aplicaciones informáticas Factores relevantes”.

Se entiende que las pistas de auditorías son rastros o huella dejadas por los contenidos a lo largo de su ciclo de vida, utilizado para analizar los procedimientos y así contar con un mejor seguimiento de las acciones realizadas por el usuario y mayor seguridad en el SGD. El concepto de pista no es solo de gran utilidad para las auditorías, sino también para la administración, ya que permiten maneras más fáciles de llevar seguimiento de la información.

Enfocado a los SGD las pistas de auditorías permiten mantener un registro de todas las acciones realizadas en un sistema de gestión de documentos electrónicos por un usuario o por el mismo sistema de forma automática permitiendo detectar cualquier violación de seguridad (2). Mayra Mena plantea que los sistemas de gestión de documentos de archivo deberán guardar las fechas, horas, identificadores únicos de los usuarios y las acciones que estos desarrollan en el sistema.

1.2 Normas y recomendaciones sobre sistemas de gestión documental y las pistas de auditorías

Los sistemas de gestión documental basan su funcionamiento en un conjunto de estándares y normas para una correcta y eficiente gestión de la información. En la investigación se realiza un estudio de algunas de estas normas y estándares para realizar un análisis de los principales planteamientos y requisitos que exponen.

ISO 15489

La norma ISO 15489 constituye la primera norma internacional en el campo de la gestión documental. Fue lanzada en su primera edición en septiembre de 2001. Esta se centra en los principios de la gestión de documentos y establece requisitos básicos para que las organizaciones puedan construir un marco de buenas prácticas que mejore de forma sistemática y efectiva la creación y mantenimiento de sus documentos apoyando las políticas y objetivos de la institución.

Esta norma define entre sus requisitos que los sistemas de gestión de documentos deben contener referencias completas y precisas de todas las operaciones que se desarrollan en relación con un documento concreto. Tales detalles pueden documentarse como parte de los metadatos¹⁵ incrustados en un documento específico, adjuntos o asociados al mismo. Alternativamente pueden registrarse como pistas de auditorías que deberían conservarse al menos mientras se conserve el documento con el que se relacionan. Los sistemas deben proporcionar pistas de auditorías u otros métodos de seguimiento con objeto de controlar las acciones que se hayan realizado (4).

¹⁵ Datos que describen el contexto, el contenido y la estructura de los documentos de archivo y su gestión a lo largo del tiempo.

Especificación MoReq

MoReq es un modelo de requisitos cuyo objetivo es definir un conjunto de consideraciones a tomar en cuenta para lograr un mejor funcionamiento de los sistemas de gestión documental electrónicos. Se diseñó para poder ser utilizado en cualquier entidad que desee implantar un sistema de gestión documental. Plantea temas referentes a la gestión documental tradicional y electrónica, además de requisitos funcionales y no funcionales que debe cumplir un sistema de gestión documental. MoReq plantea el desarrollo de un sistema de gestión documental fácil y sencillo de utilizar, con un alto grado de confianza y con la integridad deseada. Sin embargo, no se trata de una norma, los requisitos que plantea no son obligatorios, solo son recomendaciones que pueden o no ser utilizados en los sistemas de gestión documental.

Enfocado a las pistas de auditorías, MoReq plantea que todo sistema debe tener una herramienta o método capaz de registrar todas las acciones sobre los documentos, siendo llamado registro o pistas de auditorías. Estas pistas de auditorías deben ser inalterables para mantener la seguridad y confiabilidad de los datos que estas muestran, así como mostrar todas las acciones realizadas a los documentos y expedientes, la fecha y hora de estas acciones, almacenándolas todas de forma automática. Debe permitir señalar en las pistas de auditorías todas las modificaciones que afectan a grupos o documentos y los metadatos relativos a cualquiera de los elementos anteriores. Además, también debe ser capaz de generar informes sobre las acciones que afecta a los documentos y expedientes de manera organizada para un mejor análisis de toda esta información (2).

1.3 Pistas de auditorías en Alfresco

Alfresco es un Gestor de Contenidos Empresarial (ECM) de código abierto, desarrollado en 2005 por John Newton y John Powell. Alfresco presenta dos versiones, *Community* y *Enterprise*. La versión *Enterprise* incluye la mayoría de las funcionalidades de la versión *Community*, menos aquellas que se consideren experimentales, además la versión *Community*, no brinda beneficios de garantía ni soporte. Sin embargo, la versión *Enterprise* presenta licencia privativa, por lo que es necesario pagar para su utilización, mientras que la versión *Community* permite una correcta gestión de contenidos y ser más adecuado para desarrolladores capaces de conseguir soportes por sí mismos (17).

Alfresco permite unificar varios aspectos de la gestión y publicación de contenidos en una única solución informática. Uno de los aspectos más interesantes de Alfresco es la gestión documental que incorpora, permitiendo mantener de una manera organizada las carpetas y documentos. Permite grandes ventajas a sus usuarios al gestionar el contenido de los documentos, su ciclo de vida, manteniendo sobre ellos un control de versiones, además de supervisar sus flujos de trabajo. Otro aspecto a tomar en cuenta es su capacidad de búsqueda, pues permite realizarla incluso dentro de

los mismos documentos, lo que permite un acceso rápido a la información contenida en los documentos gestionados por Alfresco (18).

Pistas de auditoría

A partir de la versión 3.4.0 de Alfresco, el antiguo mecanismo de auditoría ha sido eliminado. Las viejas tablas de auditoría no se quitan, pero todo el código y configuración relacionados con la población de las tablas ha sido eliminado. Por lo tanto, es posible acceder a los datos de auditoría generados históricamente, pero todos los nuevos datos de auditoría son dirigidos a las nuevas tablas. Todas las consultas basadas en SQL utilizadas en el viejo mecanismo tienen que ser sustituidos por las llamadas a la API *AuditService* que es el mecanismo mediante el cual las aplicaciones pueden consultar o eliminar las pistas de auditoría registradas.

Las aplicaciones son pequeños módulos que definen cómo se asigna los datos, cómo se extrae y cómo se registran, ejemplo de esto es la aplicación de *alfresco-access* que es la encargada de mostrar las pistas de auditoría de las acciones sobre los contenidos, por ejemplo: creados, eliminado, modificados, entre otros. También está *AuditExampleLogin1* que mantiene un registro mostrando los inicios de sesión con éxitos, los fallidos y las salidas de sesión por usuario.

Para indicar qué se quiere o no registrar en las pistas de auditorías Alfresco tiene un mecanismo de filtrado que puede ser configurado en el fichero *alfresco-global.properties*, indicando qué acción y qué dato se quiere guardar (19).

Toda esta información se almacena en las tablas de la base de datos de Alfresco con prefijo *alf_audit*, ellas son (19):

- ✓ **alf_audit_model:** donde se registran los archivos de configuración de las auditorías.
- ✓ **alf_audit_application:** donde se registran las entradas de cada aplicación.
- ✓ **alf_audit_entry:** donde se registra cada llamada a *AuditComponent.recordAuditValues*.

Al analizar los aspectos antes planteados se puede decir que en Alfresco 4.2 existen los mecanismos necesarios para poder llevar un seguimiento de los contenidos, siendo posible desarrollar una solución factible para el problema planteado.

1.4 Herramientas para la visualización de las pistas de auditorías en Alfresco y otros sistemas

Hoy en día existen diversas herramientas para gestionar los procesos de las pistas de auditorías, mostrando reportes sobre las acciones realizadas en ellos. Con la finalidad de identificar tendencias actuales relacionadas con el mecanismo de obtención y la visualización de la información de las pistas de auditorías, además de determinar un grupo de características importantes que se puedan

tener en cuenta en el módulo a desarrollar, y agruparlas todas en una sola solución, se decide hacer un estudio de algunos de estos sistemas.

ADAudit Plus

Herramienta desarrollada por ManageEngine que permite monitorizar un servidor de directorio activo. ADAudit Plus monitoriza todos los cambios en un directorio y ofrece informes completos y alertas personalizables. A partir de estos se generan informes completos sobre las acciones realizadas por los administradores, lo que permite saber quién hizo qué y en qué momento. También permite obtener visibilidad sobre todos los cambios realizados en los usuarios y grupos, además de enviar alertas por correo electrónico si ocurre un cambio o evento específico (20).

MVD Certificate

Software de Gestión Integral para certificadoras y acreditadoras. Este *software* está compuesto por varios módulos integrados, dentro de ellos está un módulo de auditoría que provee facilidades, como permitirles a los auditores trabajar remotamente, vía Internet. Al crearse una auditoría se asigna al equipo de auditores y sólo ellos tendrán acceso a la información de la empresa auditada, incluyendo auditorías anteriores, garantizando así la confidencialidad de la información. El auditor responsable podrá solicitar la revisión administrativa y técnica, y el revisor podrá aprobar o rechazar la misma (21).

Estas herramientas solo se tuvieron en cuenta para la obtención de algunos de los requisitos funcionales y no a la hora de buscar tendencias actuales respecto a diseño gráfico y mecanismos utilizados en la gestión de los procesos de auditorías que realizan estos, debido principalmente que son herramientas propietarias que no están disponibles para poder ser instaladas y probadas en busca de diseños de interfaces de usuario que puedan ser de utilidad para el desarrollo de la solución.

Audit Surf

Herramienta desarrollada por Atol bajo licencia GPL, disponible solo para Alfresco 3.3 o 3.x, tanto para la versión *Community* como *Enterprise*, esta última con características más avanzadas. La aplicación brinda un conjunto de funcionalidades referentes a las pistas de auditorías, permitiendo visualizar de forma gráfica información de conexiones de usuarios, creación de usuarios, flujos de trabajo y estadísticas sobre creaciones, modificaciones y lecturas de documentos, además, presenta una vista única con información sobre actividades de contenidos tal como los más leídos y modificados, y las últimas incorporaciones y modificaciones realizadas en el sistema. También proporciona información sobre usuarios conectados y añadidos, información en tiempo real sobre el

estado del servidor y de los repositorios, además permite generar un fichero CSV¹⁶, XML¹⁷ o JSON¹⁸ con información filtrada por fechas y por indicadores. Esta información es obtenida utilizando la API de *Hibernate* de Alfresco, permitiendo acceder a los criterios de búsqueda para generar las consultas a la base de datos donde se encuentra almacenada la información deseada (22). Para una mejor comprensión de lo antes descrito en la siguiente imagen se muestra cómo esta herramienta visualiza los datos deseados. Capaz de exponer la información dividida por días, semanas, meses o años, observando la información mediante gráficas y exportándola en los diferentes formatos que permite la herramienta.

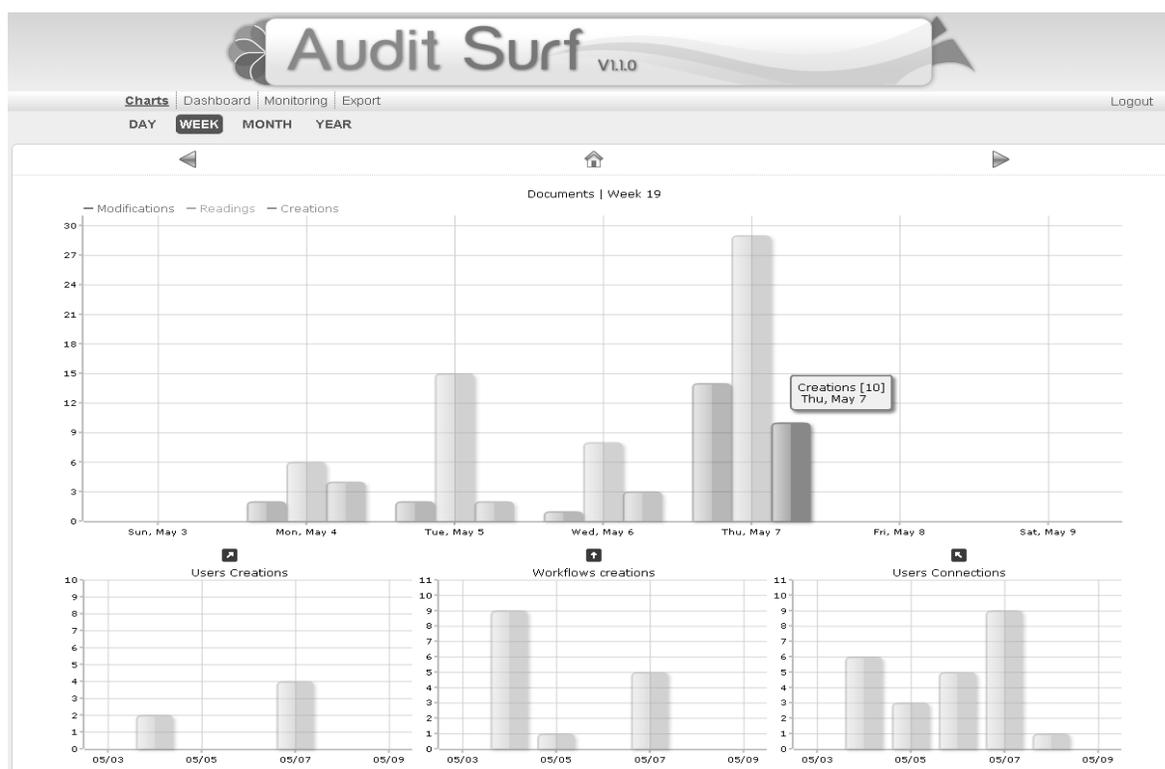


Figura 1: Visualización de las pistas de auditoría en Audit Surf.

Audit Dashlet

Audit Dashlet es una extensión que se integra con Alfresco. Esta extensión define una dashlet personalizado para mostrar los eventos de las pistas de auditorías para una aplicación dada, mostrando el identificador que corresponde la acción en la base de datos, el usuario, fecha y un

¹⁶ Acrónimo de *Comma Separated Values*, tipo de documento en formato abierto sencillo para representar datos en forma de tabla.

¹⁷ Lenguaje de Marcas eXtensible, por sus siglas en inglés (XML).

¹⁸ Acrónimo de *JavaScript Object Notation*, (Notación de Objetos de JavaScript).

campo valor (*value*) que muestra los resultados dependiendo de la aplicación seleccionada en su configuración (23). A continuación, se muestra como esta herramienta permite observar mediante una tabla un resumen de las pistas de auditorías detectadas.

DB ID	User	Timestamp	Audited Values
175	romain	7/02/13 16:09	[path : /app:company_home/st:sites/cm:test/cm:documentLibrary/cm:Pulp_Fiction-Wikipedia.pdf] [action : READ] [type : cm:content] [sub-actions : readContent] [user : romain]
174	romain	7/02/13 16:09	[path : /app:company_home/st:sites/cm:test/cm:documentLibrary/cm:Quentin_Tarantino.pdf] [action : READ] [type : cm:content] [sub-actions : readContent] [user : romain]
173	romain	7/02/13 16:09	[path : /app:company_home/st:sites/cm:test/cm:documentLibrary/cm:Albert_Einstein.pdf] [action : READ] [type : cm:content] [sub-actions : readContent] [user : romain]

Figura 2 Visualización de las pistas de auditoría en Audit Dashlet

Audit Share

Audit Share es una extensión que se integra con Alfresco. Esta aplicación muestra los documentos leídos, creados, actualizados y eliminados, mostrando la información mediante graficas divididas por día, semana y mes (24). A continuación, se muestra como esta herramienta visualiza las pistas de auditorías.



Figura 3 Visualización de las pistas de auditoría en Audit Share

Alfresco Audit Analysis and Reporting (AAAR)

Herramienta de código abierto, compatible con Alfresco. AAAR proporciona una solución para extraer, almacenar y auditar los datos junto con las informaciones de los documentos y carpetas en un nivel muy detallado. Los informes se pueden publicar en formatos (PDF¹⁹, Microsoft Excel y otros) y almacenarlos directamente en Alfresco en documentos organizados en carpetas. La aplicación permite mostrar quién está utilizando el Alfresco ECM, el crecimiento del repositorio, cuáles son los contenidos más vistos por rango de fecha, informa de los usuarios más activos por rango de fecha y muestra cuántos fallos de inicio de sesión existen en el sistema. La aplicación integra para su instalación, configuración y correcto funcionamiento las herramientas *Pentaho Data Integration* y *Pentaho Report Designer*, los cuales extraen los datos de auditoría de Alfresco y crean los informes correspondientes. Sin embargo, estas herramientas deben ser ejecutadas periódicamente para poder actualizar la información que se muestra en los reportes, siendo necesario actualizarlas en la mayoría

¹⁹ Acrónimo de *Portable Document Format* (formato de documento portátil).

de los casos una o varias veces al día (25). Debido a que algunas empresas no cuentan con un personal correctamente capacitado en el uso de las herramientas *Pentaho*, no se considera recomendable la utilización del AAAR como una solución al problema planteado. A continuación, se muestra como AAAR visualiza las pistas de auditorías permitiendo observar las acciones realizadas en rango de fechas divididas por días u horas, se visualiza la cantidad de actividades que ha realizado cada usuario, además, del tipo de acción que realizó.

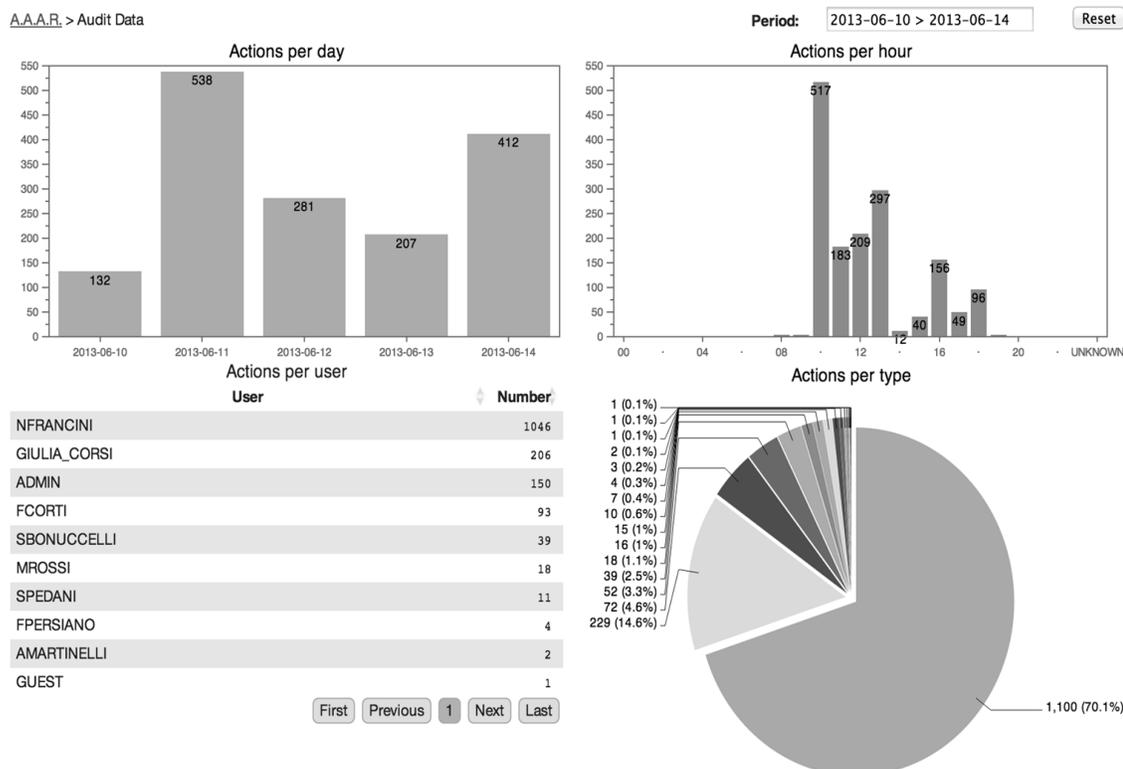


Figura 4: Visualización de las pistas de auditorías en AAAR.

Formtek Auditing Extension

Extensión desarrollada por la corporación Formtek bajo licencia privativa, siendo necesario pagar para su utilización (26). Capaz de integrarse con la aplicación Alfresco Share, dentro de sus funcionalidades tiene la de proporcionar pistas de auditoría de las acciones realizadas en los documentos, carpetas y otros tipos de contenidos almacenados en el repositorio de Alfresco. La extensión muestra información de todas las acciones que se realizaron, la fecha y la hora en que ocurrió y el usuario que la realizó. Cada carpeta y documento mantiene su propia información, posibilitando que estas pistas puedan ser vistas por cualquier usuario que tenga acceso a los documentos o carpetas, pero ninguno de ellos puede alterarlas. Para realizar las auditorías, Formtek Auditing Extension posee una herramienta, *Audit History Tool* (Herramienta de Historial de Auditoría),

que permite buscar en el repositorio las pistas de auditorías basado en diversos criterios de búsqueda, permitiendo visualizar los más recientes o todos los disponibles (27). A continuación, se muestra como Formtek Auditing Extension visualiza las pistas de auditorías mediante una tabla dividida en varias columnas, se muestra la fecha y el tipo de acción que se realizó, además del usuario que la llevo a cabo y el nombre del contenido que se ha modificado.

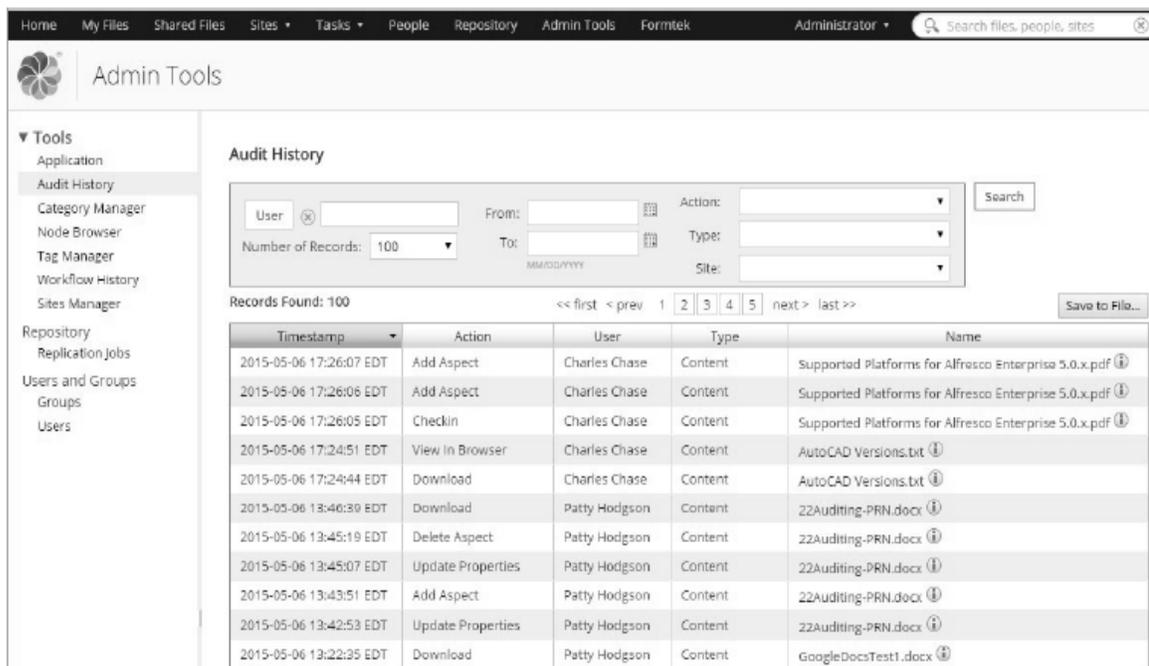


Figura 5: Visualización de las pistas de auditorías en Formtek Auditing Extension.

Con el estudio y análisis de las soluciones existentes, se obtienen los elementos que son utilizados por las pistas de auditorías con mayor frecuencia en estos sistemas, como son el nombre y el tipo de documento, las acciones que se realizan sobre estos, el usuario que la realizó y la fecha en que se ejecutó. También se identifican las diversas formas en que se puede visualizar esta información, como pueden ser tablas y gráficas que proporcionan un mejor entendimiento de la información que se quiere transmitir al usuario.

1.5 Gestor de Documentos Administrativos Xabal eXcriba

El Gestor de Documentos Administrativos eXcriba es un conjunto de tecnologías enfocadas a la gestión documental basadas en Alfresco. El objetivo principal del producto es automatizar los procesos documentales y archivísticos que se ejecutan dentro de cualquier entidad, desde la elaboración de un documento en su fase de inicio hasta su conservación o expurgo, incluyendo todas las acciones sobre los documentos tales como: crear, clasificar, control de versiones, definir tipologías

documentales, almacenar documentos en diferentes formatos electrónicos, gestionar los flujos de los documentos que se generan o reciben, salvaguardar el patrimonio documental, entre otras.

El sistema guarda las pistas de auditorías de las acciones que se realizan en él, sin embargo, solo muestra al usuario las actividades que realiza y las tareas que tiene pendiente.

En junio del 2012 fue presentado en la UCI la tesis “*Solución para los reportes de auditorías en el sistema Gestor de Documentos Administrativos eXcriba*”. La investigación, desarrollada para el eXcriba versión 2.0, propuso el desarrollo de un módulo de auditoría que permitía no solo conocer las acciones que se habían realizado sobre los documentos en el *software*, también los inicios de sesiones de un usuario, además de las modificaciones de permisos que eran realizadas sobre un documento. Esta investigación, a pesar que ayudó al proyecto eXcriba a mejorar los reportes de auditorías, no puede ser utilizada en la actualidad debido a los cambios de arquitectura ocurridos a partir de Alfresco 3.4.0.

Conclusiones parciales

El análisis de los conceptos fundamentales relacionados con las pistas de auditorías, gestión documental y sistemas de gestión documental, permitió un mejor entendimiento y dominio de las terminologías involucradas al problema identificado. Al estudiar las posibles soluciones que existen, se obtuvieron los elementos que son medidos con mayor frecuencia en las pistas de auditorías, así como las diversas formas en que se muestra esta información al usuario.

Características de la solución

En este capítulo se describe la metodología de desarrollo de *software* que se utiliza para el desarrollo de la solución, así como el lenguaje y la herramienta de modelado. Se realiza el modelo del dominio y la descripción de la arquitectura y el patrón de diseño. Además, se listan los requisitos funcionales y no funcionales junto con las descripciones de las historias de usuarios y el plan de iteración utilizado para el desarrollo de la aplicación.

2.1 Metodología de desarrollo de *software*

El desarrollo de *software* puede ser un proceso difícil de controlar, de ahí la importancia de contar con una metodología capaz de enseñar las actividades que deben desarrollarse, así como las personas involucradas y el papel que desempeñan.

AUP

El Proceso Unificado Ágil o en inglés *Agile Unified Process* (AUP) es una metodología de desarrollo de *software* simplificada del Procesos Unificado de Rational (RUP). AUP se preocupa especialmente de la gestión de riesgo incluyendo entre sus técnicas el Desarrollo Dirigido por Pruebas en inglés *Test Driven Development*. Propone que los elementos de alto riesgo tengan prioridad en el desarrollo y sean realizados en las primeras etapas. A continuación, se muestra una imagen donde se exponen las fases y disciplinas de esta metodología.

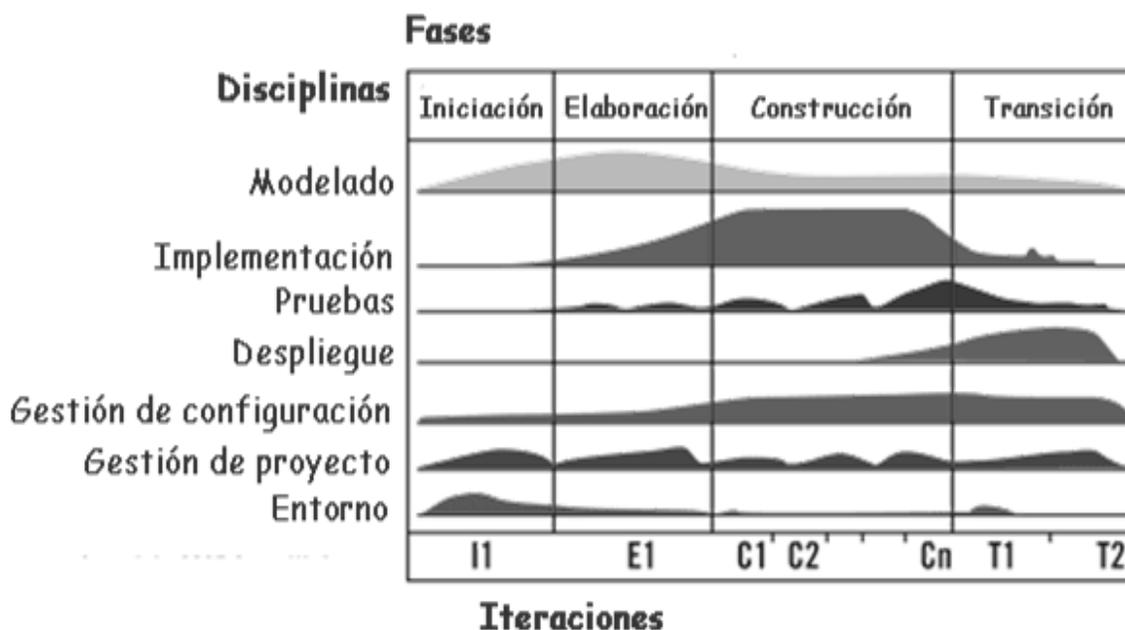


Figura 6: Fases y disciplinas de AUP.

AUP-UCI

Para el desarrollo de la solución se utiliza la variación de esta metodología desarrollada por la universidad, AUP-UCI. Debido a la variedad de metodologías utilizadas en la actividad productiva de la UCI, se decide converger a una única metodología que permita disminuir el esfuerzo en tiempo y en personas. Para esto se decide realizar una variación de la metodología AUP. Esta variación propone 3 fases para el ciclo de vida de los proyectos: Inicio, Ejecución y Cierre. Además propone 7 disciplinas, pero a un nivel más atómico que las definidas en AUP: Modelo de negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación (28). En el documento *Metodología de desarrollo para la Actividad productiva de la UCI* se realiza una descripción de las mismas:

✓ Fases de AUP-UCI:

- **Inicio:** se realizan las actividades relacionadas con la planeación del desarrollo. Se lleva a cabo un estudio del cliente con el objetivo de adquirir información que permita definir el alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo, decidiendo si se realiza el proyecto.
- **Ejecución:** se realizan las actividades necesarias para el desarrollo del *software*. Se ajusta el plan de proyecto considerando los requisitos y la arquitectura. Se obtienen los requisitos, se elabora la arquitectura y el diseño, se implementa el *software* y se libera.
- **Cierre:** en esta fase se analizan los resultados del proyecto y su ejecución, además se realizan las actividades formales de cierre de proyecto.

✓ Disciplinas de AUP-UCI:

- **Modelo de negocio:** tiene como objetivo comprender los procesos de negocio de la organización. Se investiga el funcionamiento del negocio que se desea automatizar para producir un *software* que cumpla con las necesidades de la organización. Para modelar este proceso se proponen tres variantes: Casos de Uso del Negocio, Descripción de Proceso de Negocio o Modelo Conceptual.
- **Requisitos:** se desarrolla un modelo de lo que se desea implementar. Esta fase comprende la gestión y administración de los requisitos funcionales y no funcionales. Para modelar este proceso se proponen tres variantes: Casos de Uso del Sistema, Historias de Usuario o Descripción de Requisitos por Proceso, condicionados por la variante seleccionada en el modelo de negocio.
- **Análisis y diseño:** en esta disciplina, si es necesario, se refinan los requisitos funcionales para un mayor entendimiento de estos, ayudando a una mejor

estructuración del *software*. Se modela el sistema y su estructura (arquitectura), para que soporte todos los requisitos funcionales y no funcionales.

- **Implementación:** en esta disciplina se desarrolla el *software*, siguiendo los resultados del análisis y diseño.
- **Pruebas internas:** se comprueba el resultado del desarrollo, probando las construcciones internas e intermedias, así como las versiones finales del *software*. Para esto se desarrollan artefactos de pruebas, como pueden ser Diseños de Casos de Prueba, Listas de Chequeos y Componentes de Pruebas ejecutables para automatizar estas.
- **Pruebas de liberación:** pruebas diseñadas y ejecutadas por una entidad certificada de la calidad externa. Se deben realizar a todos los entregables de proyecto, antes de ser entregados al cliente.
- **Pruebas de aceptación:** última prueba que se realiza al sistema antes de entregarlo al cliente. Es el propio cliente quien realiza esta prueba utilizando los casos de pruebas ya definidos o algunos realizados por él. Tiene como objetivo comprobar que el *software* funciona correctamente y está listo para ser utilizado por el cliente.

En el desarrollo de la solución se abarcan las disciplinas Requisitos, Análisis y diseño, Implementación y Pruebas internas. La disciplina Requisitos se aplica al realizar el levantamiento de requisitos funcionales y no funcionales, modelándolos utilizando la variante de Historias de Usuario. Al modelar el sistema mediante el diagrama de dominio, además de especificar la arquitectura que se utiliza, se abarca la disciplina de Análisis y diseño. El desarrollo de la solución cumple con lo planteado en la disciplina de Implementación y la creación de los casos de prueba abarca la disciplina Pruebas internas. No se abarca la disciplina Modelo de negocio ya que el problema identificado no presenta un negocio definido capaz de ser modelado. Además, no se abarca la disciplina de pruebas de liberación ni de aceptación.

En concordancia con las variantes seleccionadas en las disciplinas Modelado de negocio y Requisitos, AUP-UCI describe cuatro escenarios para modelar el sistema en los proyectos. Debido a que la investigación no presenta un negocio definido, se selecciona el cuarto escenario, en el cual se modela el sistema mediante Historias de Usuario.

2.2 Lenguaje de modelado UML 2.1

El lenguaje de modelado utilizado es el Lenguaje Unificado de Modelado en inglés *Unified Modeling Language* (UML), el cual se emplea para especificar, visualizar, construir y documentar artefactos de un sistema de *software*. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se emplea para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. UML proporciona un conjunto de elementos de modelado, anotaciones, relaciones y normas que pueden aplicarse a una actividad de desarrollo de *software*. Sin embargo, también puede ser empleado para modelar otros dominios, tales como el modelado de sistemas y el modelado de negocio (29). Se utiliza este lenguaje en el diseño de la solución, generando el diagrama de modelado de dominio con el objetivo de mostrar la relación que existe entre los diferentes objetos en el dominio de interés de esta investigación.

2.3 Herramienta de modelado Visual Paradigm

Existen diversas herramientas que soportan el lenguaje de modelado UML, dentro de las que se destaca *Visual Paradigm for UML* (VP-UML) por disponer de varios diagramas como los de clases, de objetos, de casos de uso del negocio y diagramas de paquetes; por la facilidad de uso para la modificación y creación de estos y por generar código a partir de dichos diagramas. Posee licencia comercial y licencia libre, aunque esta última no permite su modificación. Mediante *Visual Paradigm* versión 8.0 se puede realizar el modelado, la captura de requisitos, diseño de base de datos, modelado de procesos de negocio, entre otras funciones. *Visual Paradigm* soporta un conjunto de lenguajes, tanto en la generación del código como en la ingeniería inversa, los cuales pueden ser Java, C++²⁰, PHP²¹ y Python²², por mencionar algunos. Para maximizar la interoperabilidad de los productos de *Visual Paradigm* con otras aplicaciones se introdujo la importación y exportación de modelos de proyecto en diferentes formatos, como el XML (30).

Se utiliza el *Visual Paradigm* versión 8.0 como herramienta de modelado ya que es una herramienta multiplataforma que ofrece un ambiente favorable para realizar el análisis y diseño necesario.

2.4 Modelado de dominio

El modelo de dominio es utilizado para realizar una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Una clase conceptual puede considerarse en

²⁰ Lenguaje de programación creado a partir del lenguaje C permitiendo la manipulación de objetos y la creación de nuevos tipos de datos.

²¹ Lenguaje de programación diseñado para el desarrollo web de contenido dinámico. Puede ser usado en los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

²² lenguaje de programación multiparadigma, soporta programación orientado a objetos, programación imperativa y, en menor medida, programación funcional.

términos de un símbolo (palabras o imágenes que representan una clase conceptual), intensión (definición de una clase conceptual) y extensión (conjunto de elementos a los que se aplica la clase conceptual). Puede ser representado por diagramas de clases en la notación UML, mostrando los objetos del dominio o clases conceptuales, asociaciones entre las clases conceptuales y los atributos de las clases (31). Después de realizado un estudio se concluye que el proceso comprende pocos conceptos, siendo viable realizar un modelo de dominio para un mejor entendimiento de las clases conceptuales y objetos reales que intervienen.

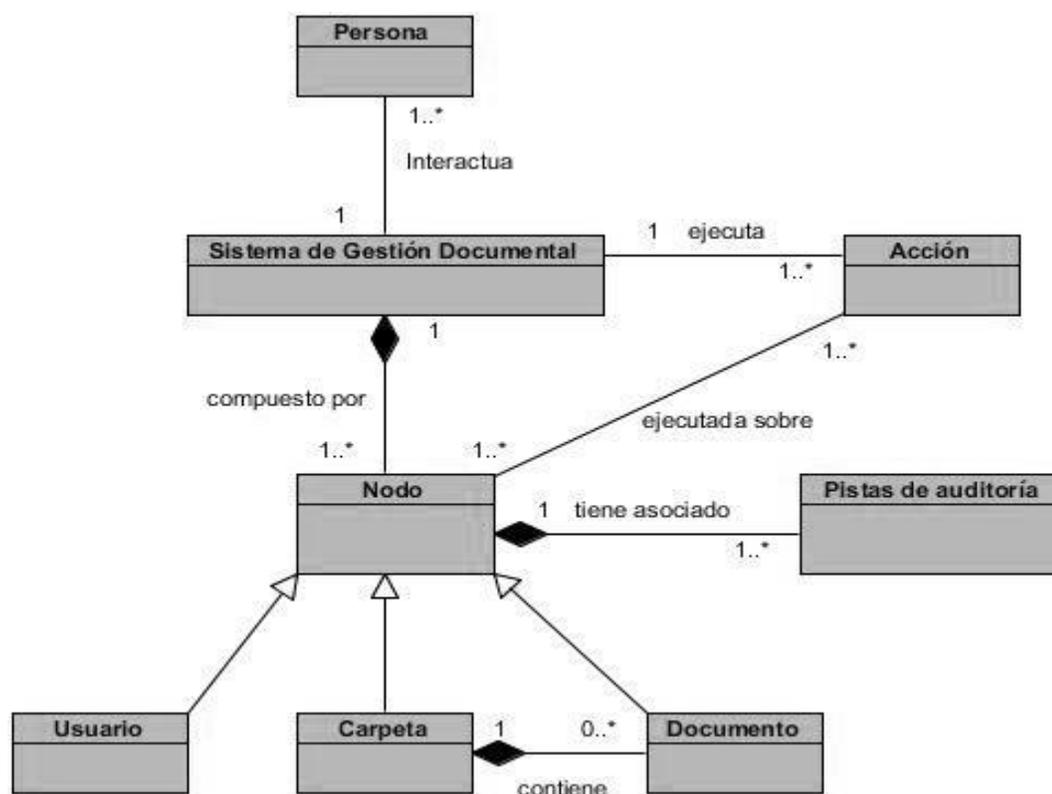


Figura 7: Modelado del Dominio.

2.4.1 Conceptos fundamentales del dominio

A continuación, se detallan cada uno de los elementos que componen el Modelo de Dominio:

- ✓ **Persona:** persona capaz de interactuar con el sistema de gestión documental y que posee los permisos para realizar diversas acciones sobre los documentos.
- ✓ **Sistema de gestión documental:** aplicación de gestión documental con la que interactúa el usuario (GDA eXcriba).
- ✓ **Acción:** Toda acción que se puede realizar en el sistema.
- ✓ **Nodo:** Objeto de un elemento en el repositorio de Alfresco que contiene atributos de un contenido (content) o de un espacio (space).

- ✓ **Pistas de auditorías:** Registros que dejan los documentos y usuarios a lo largo de su ciclo de vida.
- ✓ **Usuario:** Persona que interactuar con el sistema de gestión documental y que posee los permisos para realizar diversas en él.
- ✓ **Carpeta:** Carpeta de la que se va a mostrar las pistas de auditoría de las acciones realizadas por un usuario sobre él.
- ✓ **Documento:** Documento del que se va a mostrar las pistas de auditoría de las acciones realizadas por un usuario sobre él.

2.5 Especificación de los requisitos del software

Los requisitos muestran las necesidades requeridas por la solución propuesta, que expresan una acción o funcionalidad del mismo. Estos requisitos deben estar dirigidos a guiar el desarrollo para obtener una solución correcta y solucionar el problema planteado.

2.5.1 Requisitos funcionales

Los requisitos funcionales marcan las condiciones que la solución debe cumplir y las operaciones que se deben implementar para satisfacer las necesidades de los usuarios finales. La solución que se propone consta de los siguientes requisitos funcionales:

RF1: Mostrar las pistas de auditoría de las acciones de un usuario sobre los documentos.

RF2: Mostrar las pistas de auditoría de todos los documentos creados por un usuario.

RF3: Mostrar las pistas de auditoría de todos los documentos modificados por un usuario.

RF4: Mostrar las pistas de auditoría de todos los documentos eliminados por un usuario.

RF5: Mostrar las pistas de auditoría de todos los documentos leídos por un usuario.

RF6: Mostrar las pistas de auditoría de todos los documentos movidos por un usuario.

RF7: Mostrar las pistas de auditoría de todos los documentos copiados por un usuario.

RF8: Mostrar las pistas de auditoría de un usuario sobre los documentos en un rango de fechas.

RF9: Generar gráficas de barras de todos los documentos creados, modificados y leídos por los usuarios en el año actual.

RF10: Generar gráficas de barras de todos los documentos creados, modificados y leídos por los usuarios en el año actual agrupados por meses.

RF11: Mostrar las pistas de auditoría de todos los inicios de sesión de un usuario en el sistema.

RF12: Mostrar las pistas de auditoría de todas las salidas de sesión de un usuario en el sistema.

RF13: Mostrar las pistas de auditoría de todos los fallos de sesión de un usuario en el sistema.

RF14: Mostrar las pistas de auditoría de los registros de sesión de un usuario en el sistema en un rango de fechas.

RF15: Mostrar los usuarios que nunca han iniciado sesión en el sistema.

RF16: Mostrar las pistas de auditoría de todos los flujos de trabajo activo que ha iniciado un usuario.

RF17: Mostrar las pistas de auditoría de todos los flujos de trabajo activo que ha iniciado un usuario en un rango de fechas.

RF18: Mostrar las pistas de auditoría de todos los flujos de trabajo completado que ha iniciado un usuario.

RF19: Mostrar las pistas de auditoría de todos los flujos de trabajo completado que ha iniciado un usuario en un rango de fechas.

RF20: Mostrar las pistas de auditoría de todos los flujos de trabajo activo que han sido asignados a un usuario.

RF21: Mostrar las pistas de auditoría de todos los flujos de trabajo activo que han sido asignados a un usuario en un rango de fechas.

RF22: Mostrar las pistas de auditoría de todos los flujos de trabajo completado que han sido asignados a un usuario.

RF23: Mostrar las pistas de auditoría de todos los flujos de trabajo completado que han sido asignados a un usuario en un rango de fechas.

RF24: Mostrar las pistas de auditoría de todos los flujos de trabajo en estado activo.

RF25: Mostrar las pistas de auditoría de todos los flujos de trabajo en estado activo en un rango de fechas.

RF26: Mostrar las pistas de auditoría de todos los flujos de trabajo en estado completado.

RF27: Mostrar las pistas de auditoría de todos los flujos de trabajo en estado completado en un rango de fechas.

RF28: Imprimir los resultados de búsqueda de todas las pistas de auditoría.

2.5.2 Requisitos no funcionales

Los requisitos no funcionales representan características o cualidades que el sistema debe poseer. Estas propiedades son las que convierten al producto en algo atractivo, usable, rápido y confiable. La solución que se propone cuenta con los siguientes requisitos no funcionales:

✓ **Usabilidad**

- El módulo puede ser utilizado por personas con conocimientos básicos en el manejo de computadoras. Se utilizarán diversas palabras que permitan al usuario saber dónde debe buscar la información deseada.

✓ **Restricciones del diseño e implementación**

- Lenguaje de programación API de JavaScript de Alfresco.

✓ **Apariencia o Interfaz externa**

- Las interfaces de usuarios deben ajustarse a lo definido en el Manual de Identidad de la Universidad de las Ciencias Informáticas.

✓ **Seguridad**

- Solo el administrador debe tener acceso a la visualización de la información de las pistas de auditorías.

2.6 Actores del sistema

Un actor del sistema representa una entidad externa al sistema que interactúa con él, provocando eventos de entrada o recibiendo algo (32). A continuación, se muestra el actor y el objetivo de éste en el sistema:

Actor	Objetivo
Usuario	Representa al administrador del sistema. Es el que realiza todas las peticiones de las pistas de auditorías.

Tabla 1: Actor del sistema.

2.7 Historias de Usuario

Las historias de usuario (HU en lo adelante) son utilizadas para sustituir a los documentos de especificación funcional y a los casos de uso. Son descripciones cortas y simples de las funcionalidades del sistema, narradas desde la perspectiva de las personas que desean dichas funcionalidades, generalmente un usuario, abarcando en ocasiones múltiples funcionalidades. Las HU se centran en la comunicación entre el cliente y el equipo de desarrollo logrando documentar los requisitos del software de manera entendible para ambos, permitiendo obtener un producto lo más

cercano posible a lo que el cliente necesita (33). A continuación, se muestran tres HU que representan las funcionalidades críticas a implementar en la propuesta de solución. El resto de las HU se encuentran en el Anexo 1.

Número: 01	Nombre del requisito: Mostrar las pistas de auditoría de las acciones de un usuario sobre los documentos.
Programador: Reynaldo R.L.	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 48 horas
Riesgo en Desarrollo: N/A	Tiempo Real: 72 horas
<p>Descripción: En el dashlet Pistas de auditorías del Panel de inicio del usuario, el administrador selecciona el filtro “Historial de contenidos”. Se muestra un campo “Opción”, seleccionando en este la opción “Por usuario”. También se muestra en el <i>dashlet</i> otros campos: “Usuario”, “Acción”, “Fecha inicio”, “Fecha fin” y “Cant resultados”. En el campo “Acción” se selecciona la opción “Todas” si se desea conocer todas las acciones realizadas por el usuario. En los campos “Fecha inicio” y “Fecha fin”, se selecciona el rango de fechas deseado. En el campo “Usuario” se selecciona el usuario del que se desea conocer la información. En el campo “Cant resultados” se selecciona la cantidad máxima de elementos que desea obtener. Al oprimir el botón “Buscar”, se muestra en una tabla todas las pistas de auditorías que se han registrado del usuario seleccionado. De cada pista se detalla: el nombre del documento, el tipo de acción, la ubicación del documento y la fecha de la acción.</p>	
<p>Observaciones: El usuario debe estar autenticado en el sistema y poseer los privilegios para realizar esta acción, además debe existir en el sistema el usuario del que se desea conocer la información.</p>	
<p>Prototipo elemental de interfaz gráfica de usuario:</p>	

Pistas de auditorías

Historial de contenidos ▼

Opción:

Usuario (*):

Cant. resultados:

Acción:

Fecha inicio:

Fecha fin:

#	Nombre	Tipo	Acción	Ubicación	Fecha
1	alf docs.txt	contenido	updateNodeProperties	/app:company_home	2016-06-01 11:04:04
2	alf docs.txt	contenido	Leído	/app:company_home	2016-06-01 11:03:47
3	alf docs.txt	contenido	updateNodeProperties	/app:company_home	2016-06-01 11:02:52
4	mi doc plano	contenido	Versionado (2.0)	/app:company_home/mi carpeta de prueba de tesis	2016-06-01 09:57:10
5	mi doc plano	contenido	Creado	/app:company_home/mi carpeta de prueba de tesis	2016-06-01 09:45:55

Tabla 2: HU #01 “Mostrar todas las acciones de un usuario sobre los documentos”.

Número: 11	Nombre del requisito: Mostrar las pistas de auditoría de todos los inicios de sesión de un usuario en el sistema.
Programador: Reynaldo R.L.	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 48 horas
Riesgo en Desarrollo: N/A	Tiempo Real: 48 horas
<p>Descripción: En el dashlet Pistas de auditorías del Panel de inicio del usuario, el administrador selecciona el filtro “Historial de sesión”. Se muestra un campo “Opción”, seleccionando en este la opción “Por usuario”. También, se muestra en el <i>dashlet</i> otros campos: “Usuario”, “Acción”, “Fecha inicio”, “Fecha fin” y “Cant resultados”. En el campo “Usuario” se selecciona el usuario del que se desea conocer los inicios de sesión en el sistema. En el campo “Acción” se selecciona la opción “Inicio de sesión”. En los campos “Fecha inicio” y “Fecha fin”, se selecciona el rango de fechas deseado. En el campo “Cant resultados” se selecciona la cantidad máxima de elementos que desea obtener. Al oprimir el botón “Buscar”, se muestra en una tabla todas las pistas de auditorías sobre los inicios de sesión que se han registrado del usuario</p>	

seleccionado. De cada pista se detalla: la acción y la fecha en que se realizó.

Observaciones: El usuario debe estar autenticado en el sistema y poseer los privilegios para realizar esta acción, además debe existir en el sistema el usuario del que se desea conocer la información.

Prototipo elemental de interfaz gráfica de usuario:

#	Acción	Fecha
1	Autenticado	2016-06-02 23:19:35
2	Autenticado	2016-06-01 11:16:51
3	Autenticado	2016-06-01 11:16:47
4	Autenticado	2016-06-01 11:04:25
5	Autenticado	2016-06-01 11:04:19

Tabla 3: HU #11 “Mostrar todos los registros de sesión de un usuario en el sistema”.

Número: 16	Nombre del requisito: Mostrar las pistas de auditoría de todos los flujos de trabajo activo que ha iniciado un usuario.
Programador: Reynaldo R.L.	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 48 horas
Riesgo en Desarrollo: N/A	Tiempo Real: 48 horas
<p>Descripción: En el dashlet Pistas de auditorías del Panel de inicio del usuario, el administrador selecciona el filtro “Historial de flujos de trabajo”. Se muestra un campo “Opción”, seleccionando en este la opción “Por iniciador”. También se muestran los campos: “Usuario”, “Estado”, “Cant resultados”, “Fecha inicio” y “Fecha fin”. En el campo “Usuario” se selecciona el usuario del que se desea conocer los flujos de</p>	

trabajo que ha iniciado. En el campo “Estado” se selecciona la opción “Activos”. En los campos “Fecha inicio” y “Fecha fin”, se selecciona el rango de fechas deseado. En el campo “Cant resultados” se selecciona la cantidad máxima de elementos que desea obtener. Al oprimir el botón “Buscar”, se muestra en una tabla todas las pistas de auditorías sobre los flujos de trabajo en estado activo que han sido iniciados por el usuario seleccionado. De cada pista se detalla: el título del flujo de trabajo, el estado, la prioridad, la fecha de inicio, la fecha de vencimiento y la imagen del diagrama de modelado asociado al flujo.

Observaciones: El usuario debe estar autenticado en el sistema y poseer los privilegios para realizar esta acción, además debe existir en el sistema el usuario del que se desea conocer la información.

Prototipo elemental de interfaz gráfica de usuario:

The screenshot shows a web interface titled "Pistas de auditorías". At the top, there is a dropdown menu labeled "Historial de flujos de trabajo". Below this, there are two main search sections. The left section has a dropdown for "Opción" (set to "Por iniciador") and a text input for "Usuario (*)" (set to "admin"), with a "Buscar" button. The right section has a dropdown for "Estado" (set to "Activos"), and two text inputs for "Fecha inicio" and "Fecha fin". At the bottom right of the search area is an "Imprimir resultado" button. Below the search area is a table with the following data:

#	Titulo	Estado	Prioridad	Fecha inicio	Diagrama
1	Revisión y aprobación en conjunto	Activo	Media	2016-05-17 14:22:31	ver imagen
2	Enviar documentos para revisar	Activo	Alta	2016-05-13 10:27:06	ver imagen

Tabla 4: HU #16 “Mostrar los flujos de trabajos por el usuario iniciador en estado activos”.

2.8 Plan de iteraciones

Tomando como base cada una de las historias de usuario, se decide para su implementación fragmentar el trabajo en distintas iteraciones obteniendo un trabajo incremental. Al terminar cada iteración la solución debe tener implementadas las funcionalidades correspondientes a esta iteración. A continuación, se especifica la planificación de las HU que se implementaron en cada iteración:

Iteración 1: Esta iteración debe tener una duración total de 2 semanas.

HU #01: Mostrar las pistas de auditoría de las acciones de un usuario sobre los documentos.

HU #02: Mostrar las pistas de auditoría de todos los documentos creados por un usuario.

HU #03: Mostrar las pistas de auditoría de todos los documentos modificados por un usuario.

HU #04: Mostrar las pistas de auditoría de todos los documentos eliminados por un usuario.

HU #05: Mostrar las pistas de auditoría de todos los documentos leídos por un usuario.

HU #06: Mostrar las pistas de auditoría de todos los documentos movidos por un usuario.

HU #07: Mostrar las pistas de auditoría de todos los documentos copiados por un usuario.

HU #08: Mostrar las pistas de auditoría de un usuario sobre los documentos en un rango de fechas.

HU #09: Generar gráficas de barras de todos los documentos creados, modificados y leídos por los usuarios en el año actual.

HU #10: Generar gráficas de barras de todos los documentos creados, modificados y leídos por los usuarios en el año actual agrupados por meses.

Iteración 2: Esta iteración debe tener una duración total de 3 semanas.

HU #11: Mostrar las pistas de auditoría de todos los inicios de sesión de un usuario en el sistema.

HU #12: Mostrar las pistas de auditoría de todas las salidas de sesión de un usuario en el sistema.

HU #13: Mostrar las pistas de auditoría de todos los fallos de sesión de un usuario en el sistema.

HU #14: Mostrar las pistas de auditoría de los registros de sesión de un usuario en el sistema en un rango de fechas.

HU #15: Mostrar los usuarios que nunca han iniciado sesión en el sistema.

HU #16: Mostrar las pistas de auditoría de todos los flujos de trabajo activo que ha iniciado un usuario.

HU #17: Mostrar las pistas de auditoría de todos los flujos de trabajo activo que ha iniciado un usuario en un rango de fechas.

HU #18: Mostrar las pistas de auditoría de todos los flujos de trabajo completado que ha iniciado un usuario.

HU #19: Mostrar las pistas de auditoría de todos los flujos de trabajo completado que ha iniciado un usuario en un rango de fechas.

Iteración 3: Esta iteración debe tener una duración total de 3 semanas.

HU #20: Mostrar las pistas de auditoría de todos los flujos de trabajo activo que han sido asignados a un usuario.

HU #21: Mostrar las pistas de auditoría de todos los flujos de trabajo activo que han sido asignados a un usuario en un rango de fechas.

HU #22: Mostrar las pistas de auditoría de todos los flujos de trabajo completado que han sido asignados a un usuario.

HU #23: Mostrar las pistas de auditoría de todos los flujos de trabajo completado que han sido asignados a un usuario en un rango de fechas.

HU #24: Mostrar las pistas de auditoría de todos los flujos de trabajo en estado activo.

HU #25: Mostrar las pistas de auditoría de todos los flujos de trabajo en estado activo en un rango de fechas.

HU #26: Mostrar las pistas de auditoría de todos los flujos de trabajo en estado completado.

HU #27: Mostrar las pistas de auditoría de todos los flujos de trabajo en estado completado en un rango de fechas.

2.9 Arquitectura de la solución

La arquitectura de *software* comprende un conjunto de decisiones sobre la organización del sistema, la selección de sus elementos estructurales y sus interfaces, así como el comportamiento entre esos elementos. Es una vista estructural de alto nivel que proporciona una guía organizada para la construcción del *software*, permitiendo un mejor entendimiento entre los miembros del equipo de desarrollo (32).

2.9.1 Arquitectura en capas

La arquitectura en capas se define como una organización jerárquica tal que la mayoría de las interacciones entre los servicios solo ocurren entre capas vecinas. Esta arquitectura separa de forma clara la funcionalidad de cada capa. Además, las capas inferiores no tienen dependencia de las capas superiores permitiendo un mejor mantenimiento del código fuente, ya que si se realizan cambios en alguna de las capas no es necesario realizar cambios en las restantes (34). Su objetivo principal consiste en separar la lógica de negocio de la lógica de diseño (35). El GDA eXcriba es una aplicación donde se reconoce el estilo arquitectónico n-capas, utilizando como patrón 3-capas, donde se definen las capas: Capa de presentación, Capa Web y Capa de repositorio. Debido a que la solución propuesta se despliega como un módulo dentro del sistema, es necesario ajustarse al patrón

arquitectónico que este utiliza. A continuación, se describen las mismas y su utilización en el desarrollo de la solución:

- **Capa de presentación:** capa encargada de la interacción con el usuario y donde se visualizan todos los componentes de la *interfaz* de usuario. Es la capa donde se diseña el *dashlet* que permite al usuario introducir los datos necesarios para las peticiones a realizar y donde se visualizan las respuestas a estas peticiones, con la ayuda de la librería de JQuery y JQuery UI utilizando la API de JavaScript de Alfresco.
- **Capa web:** es la capa que contiene los métodos que son llamados desde la capa de presentación y la que accede a la capa de repositorio. Recibe toda la información y solicitudes que son introducida en la capa de presentación, realiza las peticiones al repositorio, y devuelve los resultados a la capa de presentación para presentar los resultados de las solicitudes. En esta capa se encuentran las clases que controlan el flujo de la información.
- **Capa de repositorio:** es la capa más compleja y extensa dentro del eXcriba, en ella se implementan un conjunto de servicios que permiten manipular de forma distribuida y a través de la red la información. Esta capa se implementa encima de la API remota que brinda el repositorio de Alfresco, y es la encargada de interactuar directamente con este. Es la capa encargada de acceder a los datos que el usuario solicita, y a la vez de almacenarlos. Es a la que se conecta la capa web con la intención de obtener los datos que son solicitados por el usuario en la capa de presentación.

2.10 Patrón de diseño

Los patrones de diseño en el desarrollo de *software* no son más que descripciones de problemas que se presentan continuamente, describiendo la solución base, permitiendo su utilización en varias ocasiones. Estos patrones permiten capturar las prácticas de expertos, comunicando este conocimiento a otros para resolver problemas que ya se han presentado (36).

2.10.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. GRASP es un acrónimo para *General Responsibility Assignment Software Patterns* (patrones generales de software para asignar responsabilidades). Los patrones GRASP constituyen una ayuda para entender el diseño de objetos, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Dentro de los patrones GRASP se pueden destacar los cinco principales: Experto en información, Creador, Alta Cohesión, Bajo

Acoplamiento y Controlador (32). A continuación, se mencionan los utilizados en la solución y una pequeña descripción de cada uno.

- **Experto en información:** este patrón plantea asignar la responsabilidad a la clase que contiene la información necesaria para realizar dicha responsabilidad. Si esta asignación de responsabilidades se realiza de manera correcta los sistemas tienden a ser más fáciles de entender, mantener y ampliar (32). Esto puede verse en la implementación de los web scripts haciendo cada interfaz experta según el método http que se utilice, por ejemplo, GET sin necesidad de enviar un cuerpo de petición.
- **Controlador:** este patrón asigna la responsabilidad de recibir o manejar los eventos de entrada en el sistema generados por un actor externo y definir el método como respuesta a este evento (32). Algunos de los componentes que utiliza eXcriba integran el patrón Modelo-Vista-Controlador, lo cual facilita localizar objetos responsables de ejercer la función controladora. Por ejemplo, los ficheros .js usando el lenguaje JavaScript.
- **Bajo acoplamiento:** este patrón propone asignar la responsabilidad de manera que el acoplamiento permanezca bajo. El acoplamiento es una medida de la fuerza con la que un elemento está conectado a otros. Un elemento con bajo acoplamiento no depende de demasiados elementos (32).
- **Alta cohesión:** este patrón propone asignar la responsabilidad de manera que la cohesión permanezca alta. La cohesión es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento. Un elemento con responsabilidades altamente relacionadas y que no hace una gran cantidad de trabajo, tiene alta cohesión (32).

Conclusiones Parciales

En este capítulo se logró un mejor entendimiento de como realizar la implementación de la solución. Al definir los requisitos funcionales y no funcionales se representan las características físicas y lógicas que debe presentar la solución para cumplir con su objetivo. Además, al realizar la planificación de las iteraciones durante el proceso de desarrollo del módulo, proporciona una mejor organización del trabajo. El análisis de la arquitectura y los patrones de diseño permitió conocer cuales eran los utilizados por el sistema, permitiendo lograr una estandarización en el proceso de integración.

Implementación y validación de la solución

En este capítulo se describen las herramientas, lenguajes y tecnologías utilizadas en la implementación de la solución. Se define la propuesta de solución al problema planteado. Además, se describe la validación del módulo propuesto como solución, mostrando el resultado de la prueba realizada.

3.1 Descripción de las herramientas, técnicas y lenguajes utilizados

Una correcta selección de las herramientas y lenguajes de programación permiten en algunas ocasiones facilitar el trabajo de los programadores. Los lenguajes de programación utilizados permitieron realizar las interfaces con las que interactúa el usuario, así como la obtención de las pistas de auditoría contenidas en el repositorio. Para trabajar con estos lenguajes se utiliza el editor de código Visual Studio Code.

3.1.1 JavaScript versión 1.6

JavaScript es un lenguaje orientado a objetos y multiplataforma, desarrollado por Netscape. Es más conocido como el lenguaje de *script*²³ para páginas web. Contiene una biblioteca estándar de objetos, tales como arreglos y tratamiento con fechas, permitiendo facilidades para su uso. La sintaxis básica es similar a los lenguajes de Java y C++. Las construcciones del lenguaje, tales como sentencias *if*, y bucles *for* y *while*, funcionan de la misma manera que en estos lenguajes. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web (37). En la solución del módulo se utiliza para dar cumplimiento a los requisitos funcionales, haciendo la construcción de las vistas y llamadas a los web script de Alfresco para la obtención de las pistas de auditorías.

3.1.2 API JavaScript para Alfresco versión 4.0

La API de JavaScript de Alfresco permite a los desarrolladores crear ficheros JavaScript permitiendo acceder al repositorio de Alfresco y modificar objetos que se encuentran en él, incluso crearlos y eliminarlos. Proporciona un modelo orientado a objetos para manejar conceptos de Alfresco tales como nodos, propiedades, asociaciones y aspectos (38). Para el desarrollo del módulo la API de JavaScript de Alfresco se utiliza para acceder al repositorio de Alfresco mediante llamadas AJAX.

3.1.3 AJAX

El término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML". Es una técnica de desarrollo web para crear aplicaciones interactivas. Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la

²³ Lenguaje de programación cuyo código se inserta dentro del documento HTML. Este código se ejecuta en el navegador al cargar la página web.

creación de un elemento intermedio entre el usuario y el servidor. AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor. Es una técnica válida para múltiples plataformas y utilizable en diferentes sistemas operativos y navegadores dados que está basada en estándares abiertos como JavaScript y *Document Object Model*(DOM). En el módulo se utiliza para hacer las llamadas a las aplicaciones de auditorías de Alfresco (39).

3.1.4 HTML versión 5

El Lenguaje de Marcado para Hipertextos (en inglés *HyperText Markup Language*) es un lenguaje de programación que los ordenadores son capaces de interpretar para la construcción básica de una página web en su representación visual. Sirve para estructurar documentos (títulos, párrafos, listas y otros), pero no describe la apariencia o el diseño de estos, sino que ofrece las herramientas necesarias para dar formato, según la capacidad del servidor web en el que se almacenan las páginas web y la capacidad del navegador, dígase resolución de la pantalla, fuentes de texto instaladas y otros. Dentro de sus ventajas sobresale la compatibilidad y facilidad que plantea su aprendizaje debido al reducido número de etiquetas en las que se apoya (40). En el desarrollo del módulo se utiliza para la construcción de los botones, campos de textos y tablas.

3.1.5 CSS versión 3

Las Hojas de Estilo en Cascada (*Cascading Style Sheets* conocido como CSS, por sus siglas en inglés) es el lenguaje utilizado para describir la presentación de documentos HTML o XML. Es uno de los lenguajes base de la Open Web y posee una especificación estandarizada por la organización W3C²⁴. CSS permite a los desarrolladores web llevar a cabo el control sobre el estilo y formato de múltiples páginas web, al mismo tiempo cualquier cambio en el estilo o formato de un elemento en las CSS, beneficiará a todas las páginas vinculadas a dichas CSS en las que aparezca ese elemento (41). En el módulo se utiliza para dar el estilo y colores más semejantes a los componentes ya existentes en eXcriba.

²⁴ La World Wide Web Consortium (W3C) es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la Web.

3.1.6 XML

XML es un macro-lenguaje para la creación de lenguajes de propósito especial. El propósito principal del lenguaje XML es el de facilitar la transferencia de datos a través de diferentes plataformas, especialmente las conectadas a Internet.

3.1.7 API Template de Alfresco

La API Template de Alfresco es de sólo lectura, diseñada para generar salidas HTML, XML, JSON²⁵, y texto usando el motor de plantillas²⁶ *FreeMarker*. Esta API usa una vista orientada a objetos del repositorio de contenidos combinándola con la plantilla (42).

FreeMarker es un motor de plantillas basado en Java que se centra en la arquitectura de *software* Modelo Vista Controlador. Esto ayuda en la separación de los diseñadores de páginas web HTML de los desarrolladores. *FreeMarker* fue creado originalmente para generar páginas HTML en entornos de aplicaciones web, pero también en otros tipos de aplicaciones. Este motor de plantillas es utilizado para generar la salida de los datos que proporcionan los servicios web, en formato JSON o HTML. La principal ventaja de este motor de plantillas es que la plantilla llevará la programación mínima necesaria para hacerla dinámica, dejando todo el peso de la programación fuera de la plantilla. (43).

3.1.8 Editor de código Visual Studio Code

Visual Studio Code es un editor de código que ofrece soporte para múltiples lenguajes de programación (Java, JavaScript, PHP, C++, HTML, entre otros). Es una herramienta gratuita y multiplataforma, por lo que puede ser ejecutado en diferentes sistemas operativos, como Linux y Windows. Permite a los desarrolladores la creación de servicios, aplicaciones y sitios web. Presenta varias facilidades para su uso como pueden ser los cursores múltiples o el guardado automático, además de contar con mecanismos para autocompletar códigos, aumentando la velocidad y eficiencia del programador. Una de sus características más importantes es la depuración de código permitiendo identificar errores resaltándolo en la barra de desplazamiento y resaltando la sintaxis donde se comete el error (44). Por las facilidades antes mencionadas y por la extensión implementada por el arquitecto del proyecto eXcriba que permite facilitar el completamiento de funciones y objetos de la API de JavaScript de Alfresco, es que se escoge para la implementación del módulo propuesto.

²⁵ JSON, acrónimo de JavaScript Object Notation, es un formato para el intercambio de datos.

²⁶ Una plantilla de presentación es un documento de texto que, aplicado a un modelo de datos produce una salida (un nuevo documento) que tiene la estructura definida en la plantilla.

3.2 Bibliotecas utilizadas

Las bibliotecas o librerías de programación son una colección de herramientas para la implementación de funcionalidades específicas, codificadas en un lenguaje de programación. A continuación, se describen las utilizadas en la elaboración de la solución.

3.2.1 JQuery (versión 2.2.3)

Es una biblioteca de *JavaScript*, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a las páginas web. Es *software* libre y de código abierto permitiendo su uso en proyectos libres y privados. Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en *JavaScript* que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran más resultados en un menor período de tiempo (45). En el módulo se utiliza como marco de trabajo para facilitar el uso de JavaScript.

3.2.2 JQuery UI (versión 1.8.11)

JQuery User Interface es una biblioteca de componentes para el *framework* jQuery que le añaden un conjunto de complementos y efectos visuales para la creación de aplicaciones web. Cada componente o módulo se desarrolla de acuerdo a la filosofía de jQuery (*find something, manipulate it: encuentra algo, manipúlalo*). Abstrae al programador de las implementaciones de funciones generales en navegadores particulares (46). En el presente trabajo este *framework* se utiliza para lograr un autocompletamiento en la búsqueda de usuarios registrados en el sistema, facilitando esta acción.

3.2.3 Biblioteca de gráficas amCharts versión 3.12.2

La Biblioteca gráfica amCharts es capaz de adaptarse a varias necesidades de visualización de datos. Incluye diversos tipos de gráficas, como columnas, barras, líneas, entre otras. Permite la visualización de gráficas en tres dimensiones, así como la visualización de los valores de cada eje al posicionar el *mouse* sobre ellos (47). En el módulo se utiliza para graficar todos los contenidos creados, modificados y leídos por los usuarios en eXcriba.

3.3 Propuesta del módulo

Se implementó un módulo capaz de visualizar las acciones realizadas por los usuarios sobre los documentos existentes en el sistema, así como los registros de sesiones de estos usuarios y los flujos de trabajo que son iniciados por él o asignados a este. Mediante los permisos asignados, solamente el administrador puede observar los datos correspondientes a las pistas de auditorías. En el *dashlet*, se puede seleccionar entre las opciones “Historial de contenidos”, “Historial de sesión” e “Historial de flujos de trabajo”, en dependencia de lo que se desea auditar.

En la opción “Historial de contenidos” se puede obtener los datos referentes a las acciones que se han realizado en los documentos. Permite buscar los documentos que han sido creados por un usuario, los que ha modificado o eliminado, además de los que ha leído, movido o copiado. En todas las opciones mencionadas anteriormente, se pueden realizar filtrados por fecha, los cuales selecciona el usuario. De los documentos se puede obtener el nombre, la ubicación, tipo, la acción que se realizó sobre el documento y la fecha de la acción. También permite graficar todos los documentos que han sido creados, modificados y leídos en el sistema, permitiendo agruparlos por meses en el año actual.

En la opción “Historial de sesión” se obtienen los registros de entrada, salida y fallo de sesión de un usuario en el sistema, permitiendo realizar un filtrado por fecha para la obtención de los resultados. De estos registros se muestran la acción y la fecha en que se realizó. Además, se puede obtener aquellos usuarios que nunca se han conectados al sistema. De ellos se muestra el usuario, el nombre y el correo electrónico.

En la opción “Historial de flujos de trabajo” se pueden conocer los flujos que han sido iniciados por un usuario o los que le fueron asignados a éste. De estos flujos se puede conocer el estado en que se encuentran, si están activos o ya han sido completados, seleccionando un rango de fechas deseados por el usuario para la obtención de los resultados. En caso de que el flujo se encuentre en estado activo, se muestra el nombre del flujo, el estado en que se encuentra, la prioridad que tiene el flujo, la fecha en que fue iniciado, la fecha de vencimiento y el diagrama del flujo correspondiente. En caso que se encuentre en estado completado, se muestra el nombre del flujo, el estado en que se encuentra, la prioridad que tiene el flujo, la fecha en que fue iniciado y la fecha en que concluyo. También se puede obtener los flujos de trabajo que se encuentran en estado activo o completado en todo el sistema. De los que se encuentran en estado activo se muestra el nombre del usuario iniciador, el nombre del flujo, el estado, la prioridad, la fecha en que inicio, la fecha en que vence y el diagrama de flujos correspondiente. De los que se encuentran en estado completado se muestra el nombre del usuario iniciador, el nombre del flujo, el estado, la prioridad, la fecha en que inicio y la fecha en que concluyo.

3.4 Estrategia de prueba

Las pruebas de software se consideran un elemento crítico para garantizar la calidad de software representando una revisión final de las especificaciones y el diseño. Las pruebas se realizan con la intención de encontrar el mayor número de errores posibles con la mínima cantidad de esfuerzo y tiempo posible, antes de entregar al cliente el producto final. Demuestran hasta qué punto las funciones del software funcionan correctamente, de acuerdo con los requisitos planteados. Además,

los resultados arrojados por las pruebas proporcionan una indicación de la calidad y fiabilidad del software obtenido (49). Para validar la solución propuesta se utilizan el método de caja negra.

Pruebas de caja negra

Las pruebas de caja negra se refieren a las pruebas que se llevan a cabo sobre la interfaz del *software*. Permiten obtener un conjunto de condiciones de entrada que ejerciten todos los requisitos funcionales del software. Estas pruebas permiten encontrar funciones incorrectas o ausentes, errores de interfaz, errores de acceso a datos y de rendimiento (49).

Al módulo desarrollado se le aplica la prueba de caja negra, utilizando el método de partición equivalente. Este método propone dividir los campos de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de prueba se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada (49).

A continuación, se describen las variables que representan las posibles condiciones de entrada en el sistema.

Nombre de la variable	Clasificación	Valor Nulo	Descripción
Opción	Lista de selección única	No	En este campo solo puede escogerse entre las opciones disponibles.
Usuario	Campo de texto	No	En este campo solo pueden introducirse letras.
Cant. resultados	Campo de texto	Sí	En este campo solo pueden introducirse números enteros.
Acción	Lista de selección única	No	En este campo solo puede escogerse entre las opciones disponibles.
Estado	Lista de selección única	No	En este campo solo puede escogerse entre las opciones disponibles.
Fecha inicio	Campo de fecha	Sí	En este campo se escoge la fecha en el calendario.

Fecha fin	Campo de fecha	Sí	En este campo se escoge la fecha en el calendario.
Rango de fecha	Lista de selección única	No	En este campo solo puede escogerse entre las opciones disponibles.

Tabla 5 Descripción de las variables

Después de haber definido las posibles condiciones de entrada, se describen los tres casos de pruebas críticos para la solución a partir de las tres HU definidas anteriormente. El resto de los casos de prueba se encuentran en el Anexo 2.

SC 1 Mostrar las pistas de auditoría de las acciones de un usuario sobre los documentos.

Escenario	Opción	Usuario	Acción	Cant. resultados	Fecha Inicio	Fecha Fin	Respuesta del sistema
EC 1.1 Visualizar pistas de auditoría correctamente	V	V	V	N/A	N/A	N/A	El sistema muestra en una tabla todas las pistas de auditorías que se han registrado del usuario seleccionado sobre los documentos.
	Por usuario	admin	Todas				
EC 1.2 Validar inserción de datos incorrectos.	V	I	V	N/A	N/A	N/A	El sistema muestra un mensaje indicando que existen datos incorrectos.
	Por usuario	admin+	Todas				
EC 1.3 Validar	V	I	V	N/A	N/A	N/A	El sistema

campos obligatorios.	Por usuario		Todas				muestra un mensaje indicando que hay campos vacíos.
----------------------	-------------	--	-------	--	--	--	---

Tabla 6: Descripción del caso de prueba # 01 de la HU #01.

SC 2 Mostrar las pistas de auditoría de todos los inicios de sesión de un usuario en el sistema.

Escenario	Opción	Usuario	Acción	Cant. resultados	Fecha Inicio	Fecha Fin	Respuesta del sistema
EC 2.1 Visualizar pistas de auditoría correctamente .	V	V	V	N/A	N/A	N/A	El sistema muestra en una tabla todas las pistas de auditorías sobre los inicios de sesión que se han registrado del usuario seleccionado.
	Por usuario	admin	Inicio de sesión				
EC 2.2 Validar inserción de datos incorrectos.	V	I	V	N/A	N/A	N/A	El sistema muestra un mensaje indicando que existen datos incorrectos.
	Por usuario	admin+	Inicio de sesión				
EC 2.3 Validar campos obligatorios.	V	I	V	N/A	N/A	N/A	El sistema muestra un mensaje indicando que hay campos vacíos.
	Por usuario		Inicio de sesión				

Tabla 7 Descripción del caso de prueba # 02 de la HU #11

SC 3 Mostrar las pistas de auditoría de todos los flujos de trabajo activo que ha iniciado un usuario.

Escenario	Opción	Usuario	Acción	Cant. resultados	Fecha Inicio	Fecha Fin	Respuesta del sistema
EC 3.1 Visualizar pistas de auditoría correctamente.	V	V	V	N/A	N/A	N/A	El sistema muestra en una tabla todas las pistas de auditorías sobre flujos de trabajo activo que ha iniciado un usuario.
	Por iniciador	admin	Activos				
EC 3.2 Validar inserción de datos incorrectos.	V	I	V	N/A	N/A	N/A	El sistema muestra un mensaje indicando que existen datos incorrectos.
	Por iniciador	admin+	Activos				
EC 3.3 Validar campos obligatorios.	V	I	V	N/A	N/A	N/A	El sistema muestra un mensaje indicando que hay campos vacíos.
	Por iniciador		Activos				

Tabla 8 Descripción del caso de prueba # 03 de la HU #16

Para comprobar que el resultado obtenido coincide con los resultados esperados se diseñaron 28 casos de pruebas, realizando un total de tres iteraciones por cada caso de prueba. Con la ejecución de 13 casos de pruebas, en la primera iteración, se obtuvo el resultado esperado; mientras que la aplicación de 15 casos de pruebas resultó no satisfactoria.

En esta primera iteración se detectaron un total de 16 no conformidades, de las cuales 4 se clasificaron como altas y el resto, bajas. Las no conformidades altas consistían en errores de funcionamiento en algunas de las funcionalidades críticas de la solución. Por ejemplo, en la sección “Historial de contenidos” no se realizaba un correcto filtrado por fechas, ya que mostraba acciones

sobre los documentos fuera del rango seleccionado; además, no se realizaba un correcto agrupamiento al graficar la cantidad de acciones que habían realizados los usuarios por meses. En las secciones “Historial de sesión” e “Historial de flujos de trabajo” también se detectó la no conformidad en el incorrecto filtrado por fechas. El resto de las no conformidades clasificadas como bajas se debían a errores ortográficos y además, en las gráficas no se resaltaban en diferentes colores los agrupamientos de las acciones realizadas por los usuarios, lo que atentaba contra la experiencia de usuario.

En la segunda iteración, con la ejecución de 25 casos de pruebas se obtuvo el resultado esperado, mientras que 3 pruebas resultaron fallidas. En resumen, solo se detectaron 4 no conformidades clasificadas como baja, que se debían a errores ortográficos. En la última iteración, la ejecución de los 28 casos de prueba resultó satisfactoria.

Conclusiones parciales

En este capítulo se logró un mejor entendimiento de cómo dar solución al problema que se plantea, al definir las herramientas, tecnologías y lenguajes de programación para construir la solución que se propone. Además, la realización de las pruebas de *software* al sistema permitió detectar errores existentes, solucionándolos antes de la entrega del producto al cliente.

Conclusiones generales

Al finalizar el presente trabajo se arribó a las siguientes conclusiones:

- El estudio de los mecanismos existentes para la visualización de las pistas de auditorías permitieron conocer las principales características de estos, identificando así los requisitos funcionales y no funcionales propuestos en la presente investigación para el desarrollo del módulo.
- La selección de la metodología, herramienta y lenguaje de modelado permitió el diseño de un módulo capaz de satisfacer las necesidades planteadas en el problema.
- La implementación de cada una de las funcionalidades definidas para la solución utilizando las herramientas y lenguajes de programación permitieron obtener un producto a la altura de las necesidades del proyecto eXcriba.
- La aplicación de las pruebas permitió corregir las no conformidades detectadas en el módulo de manera satisfactoria, Lo que contribuyó a la obtención de un producto final listo para ser desplegado en la versión 3.1 del GDA Xabal eXcriba.
- El módulo desarrollado posibilitó la visualización de las pistas de auditorías a nivel de usuario en el GDA Xabal eXcriba 3.1, lo que permite proveer de información oportuna a los administradores del sistema para la toma de decisiones.

La investigación propone las siguientes recomendaciones:

- Realizar pruebas de rendimientos al módulo con un gran volumen de información en un entorno real.
- Adicionar la funcionalidad que permita registrar las pistas de auditoría sobre la modificación de permisos que se han realizado sobre un documento en específico.
- Valorar la inclusión de la solución en la evolución al eXcriba 4.0.

Referencias Bibliográficas

1. **Mugica, Mayra Mena.** *Gestión documental y organización de archivos.* La Habana : Félix Varela, 2005. ISBN 959-258-950-X.
2. **MoReq.** *Modelo de requisitos para la gestión de documentos electrónicos de archivos.* 2001.
3. **Mugica, Mayra Mena, y otros.** *LA GESTIÓN DOCUMENTAL COMO SOPORTE DEL SISTEMA DE GESTIÓN DEL CONOCIMIENTO EN LAS ORGANIZACIONES CUBANAS.* La Habana : Revista Científica, 2010. ISSN 1562-3297.
4. **ISO.** *ISO 15489 Información y documentación. Gestión de documentos.* Ottawa : s.n., 2001.
5. **Tribunal Supremo de Elecciones.** *POLÍTICA INSTITUCIONAL DE GESTIÓN DOCUMENTAL.* Tribunal Supremo de Elecciones. 2014.
6. **Diccionario de Terminología Archivística.** Madrid : s.n., 1992.
7. **Tribunal Supremo de Elecciones Normativa.** *POLÍTICA INSTITUCIONAL DE GESTIÓN DOCUMENTAL.* 2014.
8. **Codina, Lluís.** *Sistemas de gestión documental: estado del arte y estrategia de utilización.* 1994.
9. **ISO 30300.** 2011.
10. **Mejía, Guillermo Adolfo Cuellar.** *Concepto universal de auditoría.* 2009.
11. **Rosa, Alberto Salinas La.** *Inteligencia de Negocio. Auditoría y control. Prototipo de herramienta de calidad.* Madrid : s.n., 2010.
12. **Troncoso, Ricardo Vilches.** *APUENTES DEL ESTUDIANTE DE AUDITORÍA.* Chile : s.n.
13. **Ellis, Charles D.** *Information audits, communication audits and information mapping: a review and summary.* s.l. : International Journal of Information Management, 1993.
14. **Buchanan, S. & Gibb, F.** *The information audit: an integrated strategic approach.* Australia : International Journal of Information Management, 1998.
15. **Castellano, Ricardo.** *Auditoría en entornos informáticos.* 2006. I.S.B.N. 950-33-0199-8.
16. **Guido, Sergio Espinosa.** *Las pistas de auditorías.* 2012.
17. **Alfresco.** Alfresco. [En línea] 2016. <https://www.alfresco.com/es/products/enterprise-content-management/community>.

18. **Borrego, Daniel.** Herramientas para PYMES. [En línea] 2009. <http://www.herramientasparapymes.com/gestor-de-contenido-alfresco-community-open-source>.
19. **Alfresco.** Alfresco Documentation. [En línea] 2016. <http://docs.alfresco.com/4.1/concepts/audit-intro.html>.
20. **ManageEngine.** Software de auditoría de cambios de Active Directory. *manageengine*. [En línea] 2016. <https://www.manageengine.es/active-directory-audit/>.
21. **MVD-TECHNOLOGIES.** MVD Certificate software de Gestión Integral para Certificadoras y Acreditadoras. [En línea] 2010. <http://www.mvdtech.com/certificate.aspx>.
22. **Ugartondo, Alejandro.** Standard Operating Procedure: Auditar con Alfresco. *Standard Operating Procedure: Auditar con Alfresco*. [En línea] 5 de enero de 2011. <http://standardoperationprocedure.blogspot.com/2011/01/auditar-con-alfresco.html>.
23. **Guinot, Romain.** Audit Application dashlet for Alfresco Share. *github*. [En línea] 2015. <https://github.com/share-extras/audit-dashlet>.
24. **changes, Minor.** "AuditShare" for Alfresco. *github*. [En línea] 2014. <https://github.com/atolcd/alfresco-audit-share>.
25. **Corti, Francesco.** Alfresco Audit Analysis and Reporting. [En línea] 2013. <http://fcorti.com/alfresco-audit-analysis-reporting/>.
26. **Alfresco.** Alfresco Add Ons. [En línea] 2016. <https://addons.alfresco.com/addons/auditing-extension>.
27. **Formtek Inc.** *Formtek Auditing Extension for Alfresco*. US : s.n., 2016.
28. **UCI.** *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : s.n., 2015.
29. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El proceso Unificado de Desarrollo de Software*. 2007.
30. **Visual Paradigm.** What is Visual Paradigm? [En línea] Visual Paradigm, 2016. <https://www.visual-paradigm.com/features/>.
31. **Larman, Craig.** *MODELO DEL DOMINIO*. s.l. : Prentice Hall, 2003.
32. **Larman, Craig.** *UML y Patrones*. Vancouver : s.n., 1999.

33. **Izaurrealde, María Paula.** *Caracterización de Especificación de Requerimientos en entornos Ágiles: Historias de Usuario.* Córdoba : Universidad Tecnológica Nacional Facultad Regional Córdoba, 2013.
34. **Torre Llorante, César de la, y otros.** *Guía de arquitectura N-capas orientada al dominio con .Net 4.0.* España : s.n., 2010. ISBN 978-84-936696-3-8.
35. **Henríquez, Santiago Domingo Moquillaza, Huerta, Hugo Vega y Grados, Luis Guerra.** *Programacion en N capas.* San Marcos : s.n., 2010.
36. **Tello, Jesús Cáceres.** *Patrones de diseño.* 2008.
37. **Gorrotowi.** MDN Mozilla Developer Network. [En línea] 2014. <https://developer.mozilla.org/es/docs/Glossary/JavaScript>.
38. **Alfresco Wiki.** Web_Scripts. *wiki.alfresco.* [En línea] 2015. https://wiki.alfresco.com/wiki/Web_Scripts.
39. **Eguiluz, Javier.** Introducción a AJAX. *Librosweb.* [En línea] 2016. <http://librosweb.es/libro/ajax/>.
40. **Vértice.** *Diseño Básico de Páginas Web en HTML.* España : Publicaciones Vértice S.L, 2009.
41. **Martínez, Rogelio.** MDN Mozilla Developer Network. [En línea] 2015. <https://developer.mozilla.org/es/docs/Web/CSS>.
42. **ematiz.** *Alfresco 4.2 para desarrolladores.* 2014.
43. **Piergiorgio Lucidi, Ugo Cei.** *Alfresco 3 Web Services.* 2010.
44. **Visual Studio.** Visual Studio Code. *Visual Studio Code.* [En línea] 2016. <https://www.visualstudio.com/es-es/products/code-vs.aspx>.
45. **Portal JQuery.** JQuery. [En línea] 2016. <http://jquery.com/>.
46. **Portal JQuery UI.** JQuery UI. [En línea] 2016. <http://jqueryui.com/>.
47. **amCharts.** amCharts: JavaScript Charts and Maps. [En línea] 2015. <https://www.amcharts.com/>.
48. **Alfresco Wiki.** Web_Scripts. *wiki.alfresco.* [En línea] 2015. https://wiki.alfresco.com/wiki/JavaScript_API.
49. **Pressman, Roger.** *Ingeniería de Software: un enfoque practico.* España : s.n., 2001.

Bibliografía

1. **Mugica, Mayra Mena.** *Gestión documental y organización de archivos.* La Habana : Félix Varela, 2005. ISBN 959-258-950-X.
2. **MoReq.** *Modelo de requisitos para la gestión de documentos electrónicos de archivos.* 2001.
3. **Mugica, Mayra Mena, y otros.** *LA GESTIÓN DOCUMENTAL COMO SOPORTE DEL SISTEMA DE GESTIÓN DEL CONOCIMIENTO EN LAS ORGANIZACIONES CUBANAS.* La Habana : Revista Científica, 2010. ISSN 1562-3297.
4. **ISO.** *ISO 15489 Información y documentación. Gestión de documentos.* Ottawa : s.n., 2001.
5. **Tribunal Supremo de Elecciones.** *POLÍTICA INSTITUCIONAL DE GESTIÓN DOCUMENTAL.* Tribunal Supremo de Elecciones. 2014.
6. **Diccionario de Terminología Archivística.** Madrid : s.n., 1992.
7. **Tribunal Supremo de Elecciones Normativa.** *POLÍTICA INSTITUCIONAL DE GESTIÓN DOCUMENTAL.* 2014.
8. **Codina, Lluís.** *Sistemas de gestión documental: estado del arte y estrategia de utilización.* 1994.
9. **ISO 30300.** 2011.
10. **Mejía, Guillermo Adolfo Cuellar.** *Concepto universal de auditoría.* 2009.
11. **Rosa, Alberto Salinas La.** *Inteligencia de Negocio. Auditoría y control. Prototipo de herramienta de calidad.* Madrid : s.n., 2010.
12. **Troncoso, Ricardo Vilches.** *APUENTES DEL ESTUDIANTE DE AUDITORÍA.* Chile : s.n.
13. **Ellis, Charles D.** *Information audits, communication audits and information mapping: a review and summary.* s.l. : International Journal of Information Management, 1993.
14. **Buchanan, S. & Gibb, F.** *The information audit: an integrated strategic approach.* Australia : International Journal of Information Management, 1998.
15. **Castellano, Ricardo.** *Auditoría en entornos informáticos.* 2006. I.S.B.N. 950-33-0199-8.
16. **Guido, Sergio Espinosa.** *Las pistas de auditorías.* 2012.
17. **Alfresco.** Alfresco. [En línea] 2016. <https://www.alfresco.com/es/products/enterprise-content-management/community>.

18. **Borrego, Daniel.** Herramientas para PYMES. [En línea] 2009. <http://www.herramientasparapymes.com/gestor-de-contenido-alfresco-community-open-source>.
19. **Alfresco.** Alfresco Documentation. [En línea] 2016. <http://docs.alfresco.com/4.1/concepts/audit-intro.html>.
20. **ManageEngine.** Software de auditoría de cambios de Active Directory. *manageengine*. [En línea] 2016. <https://www.manageengine.es/active-directory-audit/>.
21. **MVD-TECHNOLOGIES.** MVD Certificate software de Gestión Integral para Certificadoras y Acreditadoras. [En línea] 2010. <http://www.mvdtech.com/certificate.aspx>.
22. **Ugartondo, Alejandro.** Standard Operating Procedure: Auditar con Alfresco. *Standard Operating Procedure: Auditar con Alfresco*. [En línea] 5 de enero de 2011. <http://standardoperationprocedure.blogspot.com/2011/01/auditar-con-alfresco.html>.
23. **Guinot, Romain.** Audit Application dashlet for Alfresco Share. *github*. [En línea] 2015. <https://github.com/share-extras/audit-dashlet>.
24. **changes, Minor.** "AuditShare" for Alfresco. *github*. [En línea] 2014. <https://github.com/atolcd/alfresco-audit-share>.
25. **Corti, Francesco.** Alfresco Audit Analysis and Reporting. [En línea] 2013. <http://fcorti.com/alfresco-audit-analysis-reporting/>.
26. **Alfresco.** Alfresco Add Ons. [En línea] 2016. <https://addons.alfresco.com/addons/auditing-extension>.
27. **Formtek Inc.** *Formtek Auditing Extension for Alfresco*. US : s.n., 2016.
28. **UCI.** *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : s.n., 2015.
29. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El proceso Unificado de Desarrollo de Software*. 2007.
30. **Visual Paradigm.** What is Visual Paradigm? [En línea] Visual Paradigm, 2016. <https://www.visual-paradigm.com/features/>.
31. **Larman, Craig.** *MODELO DEL DOMINIO*. s.l. : Prentice Hall, 2003.
32. **Larman, Craig.** *UML y Patrones*. Vancouver : s.n., 1999.

33. **Izaurrealde, María Paula.** *Caracterización de Especificación de Requerimientos en entornos Ágiles: Historias de Usuario.* Córdoba : Universidad Tecnológica Nacional Facultad Regional Córdoba, 2013.
34. **Torre Llorante, César de la, y otros.** *Guía de arquitectura N-capas orientada al dominio con .Net 4.0.* España : s.n., 2010. ISBN 978-84-936696-3-8.
35. **Henríquez, Santiago Domingo Moquillaza, Huerta, Hugo Vega y Grados, Luis Guerra.** *Programacion en N capas.* San Marcos : s.n., 2010.
36. **Tello, Jesús Cáceres.** *Patrones de diseño.* 2008.
37. **Gorrotowi.** MDN Mozilla Developer Network. [En línea] 2014. <https://developer.mozilla.org/es/docs/Glossary/JavaScript>.
38. **Alfresco Wiki.** Web_Scripts. *wiki.alfresco.* [En línea] 2015. https://wiki.alfresco.com/wiki/Web_Scripts.
39. **Eguiluz, Javier.** Introducción a AJAX. *Librosweb.* [En línea] 2016. <http://librosweb.es/libro/ajax/>.
40. **Vértice.** *Diseño Básico de Páginas Web en HTML.* España : Publicaciones Vértice S.L, 2009.
41. **Martínez, Rogelio.** MDN Mozilla Developer Network. [En línea] 2015. <https://developer.mozilla.org/es/docs/Web/CSS>.
42. **ematiz.** *Alfresco 4.2 para desarrolladores.* 2014.
43. **Piergiorgio Lucidi, Ugo Cei.** *Alfresco 3 Web Services.* 2010.
44. **Visual Studio.** Visual Studio Code. *Visual Studio Code.* [En línea] 2016. <https://www.visualstudio.com/es-es/products/code-vs.aspx>.
45. **Portal JQuery.** JQuery. [En línea] 2016. <http://jquery.com/>.
46. **Portal JQuery UI.** JQuery UI. [En línea] 2016. <http://jqueryui.com/>.
47. **amCharts.** amCharts: JavaScript Charts and Maps. [En línea] 2015. <https://www.amcharts.com/>.
48. **Alfresco Wiki.** Web_Scripts. *wiki.alfresco.* [En línea] 2015. https://wiki.alfresco.com/wiki/JavaScript_API.
49. **Pressman, Roger.** *Ingeniería de Software: un enfoque practico.* España : s.n., 2001.
50. **Corti, Francesco.** fcorti. [En línea] marzo de 2013. <http://fcorti.com/alfresco-audit-analysis-reporting/>.

51. **Mejía, Guillermo Adolfo Cuéllar.** fccea.unicauca. [En línea] Universidad del Cauca, Facultad de Ciencias Contables Económicas y Administrativas, 2003. <http://fccea.unicauca.edu.co/old/tgarf/tgarfse3.html>.
52. **CGE.** cge. [En línea] 2008. http://www.cge.es/portalcge/novedades/2009/prl/pdf_auditoria/capitulo3_1.pdf.
53. **Diccionario, de la Lengua Española.** Definición de/reporte. *Definición de.* [En línea] 2016. <http://definicion.de/reporte/>.
54. **Atol.** blog.atolcd. [En línea] diciembre de 2009. <https://blog.atolcd.com/presentation-technique-dauditsurf/#more-321>.
55. **Openkm.** openkm. [En línea] 2011. <http://www.openkm.com/es/>.
56. **Tristán Elósegui.** tristanelosegui. [En línea] 2014. <http://tristanelosegui.com/2014/10/27/que-es-y-para-que-sirve-un-dashboard/>.
57. **Malik, Shadam.** *Interprise Dashboard, desing and best Practices for IT.* s.l. : John Wiley & Sons, Inc., Hoboken, New Jersey., 2005.
58. **Headway Digital.** Introducción a la analítica web- KPI's- Google Analytics - Métricas web. *Linkedin.* [Online] 2012.
59. **BI Dashboards.** *BI Dashboards.* [En línea] 2011. <http://www.bidashboard.org/>.
60. **blogspot.** TIPOS Y CARACTERISTICAS DE REPORTES. [En línea] 2010. <http://angiemilenamogollonmonsalve.blogspot.com/>.
61. **Universidad Tecnológica Latinoamericana .** Reportes. *utel university.* [En línea] 2005. www.utel.edu.mx.
62. **Casals, Ing. Velmour Muñoz.** *Sistema de Gestión Documental.* 2007.
63. **Cawthorne, Jed.** CMS or ECM – What is the difference? [En línea] 2010. <http://www.prescientdigital.com/articles/content-management>.
64. **AIIM.** What is Enterprise Content Management (ECM)? [En línea] 2010. <http://www.aiim.org/What-is-ECM-Enterprise-Content-Management>.
65. **Córdova, Laura Roque.** [En línea] 2012. <http://revistatelematica.cujae.edu.cu/index.php/tele/article/viewFile/17/11>.

66. **Athento**. Athento: Understanding Documents. [En línea] [Citado el: 15 de 03 de 2016.] <http://www.athento.com/nuxeo/caracteristicas/>.
67. —. Athento. [En línea] [Citado el: 15 de 3 de 2016.] <http://www.athento.com/caracteristicas-tecnicas-athento-iDM/>.
68. **ISO 15489**. 2001.
69. **MoReq**. *Modelo de requisitos para la gestión de documentos electrónicos de archivos*. 2001.
70. **Riola, Jose Carlos Carvajal**. *METODOLOGÍAS ÁGILES. Tesis final de master*. Barcelona : s.n., 2008.
71. **Sala, Jesus Javier Rodríguez**. *Introducción a la programación: Teoría y practica*. 2003.
72. **Netbeans**. Netbeans. [En línea] 2012. [Citado el: 16 de 03 de 2016.] http://netbeans.org/index_es.html.
73. **ISO 15489-1**. 2005.
74. **Ponjuan, Gloria**. *Sistema de Información, Principios Aplicaciones*. La Habana : s.n., 2004.
75. **Collado, Julián Moyano**. *Desarrollo, e implantación de un sistema de gestión documental en una organización*. Zaragoza : s.n., 2009.
76. **Athento**. *Anexo descriptivo de funcionalidades de Nuxeo EP 5.5*. 2015.
77. **Ponjuan, Gloria, y otros**. *SISTEMAS DE INFORMACIÓN: PRINCIPIOS Y*. Habana : s.n., 2004.
78. **Arnold, Marcelo y Osorio, Francisco**. *Introducción a los conceptos básicos de la teoría general de sistemas*. 1998.
79. **Díaz, Fernando Antonio Ávila**. *LA GESTIÓN DOCUMENTAL EN CUBA: RESULTADO DE LA POLÍTICA CULTURAL DE LA REVOLUCIÓN*. Santiago de Cuba : s.n., 2007.
80. **Fernández Romero, Yenisleidy y Díaz González, Yanette**. *Patrón Modelo-Vista-Controlador*. La Habana : s.n., 2012. ISSN 1729-3804.
81. **página oficial de Alfresco**. Auditar con Alfresco. [En línea] 2011. <https://www.alfresco.com/es/noticias/cobertura/auditar-con-alfresco>.
82. **c, cv**. *vcncvn*. 2001.
84. **Reynoso, Carlos y Kicillof, Nicolás**. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004.

85. **Universidad de Castilla.** *CURSO DE FORMACIÓN* . 2008.
86. **Sanchidrian, Yanay Viera Lorenzo y Daimara Mustelier.** *Gestión documental del Sistema de Gestión Penitenciaria.* La Habana : s.n., 2007.
87. **Castello, Ricardo J.** *Auditoría en entornos informáticos.* 2006. I.S.B.N. 950-33-0199-8.
88. **González, María de los Ángeles Ruiz y Bas, Ariel Bodes.** *La Gestión Documental y su Impacto en el Sector Empresarial Cubano.* La Habana : GECONTEC, 2014. ISSN 2255-5684.