

Implementación: parametrización y personalización de herramientas

Carlota Bustelo Ruesta

PID_00203146



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

Índice

Introducción	5
Objetivos	6
1. El plan de implementación	7
1.1. ¿Qué actividades o tareas hay que llevar a cabo?	7
1.1.1. Diseño técnico e implementación	8
1.1.2. Pruebas y validación	9
1.1.3. Acciones de gestión del cambio	11
1.2. Definición de responsabilidades	13
1.3. Documentación	13
1.4. Despliegue	15
2. Modelo de información: los objetos documentales	17
2.1. Entendiendo las opciones de la tecnología	17
2.2. La definición de los repositorios o espacios	19
2.3. La diferencia entre <i>document</i> y <i>records</i> ; el paso de <i>document</i> a <i>record</i>	23
2.3.1. Cambio de tipo de objeto	23
2.3.2. <i>File Plan</i> o clasificación	24
2.3.3. Relación entre los objetos	25
2.4. La utilización de objetos predefinidos y su personalización	26
2.5. Los tipos o clase documentales: cómo implementar los metadatos	28
2.5.1. Pilares básicos	28
2.5.2. Construcción del perfil de aplicación	28
2.5.3. Implementar metadatos no definidos en el modelo	29
2.5.4. Definición de las formas de cumplimentación	30
2.5.5. Documentar el perfil de aplicación	31
3. La implementación del modelo de acceso	32
3.1. Consideraciones generales	32
3.2. Autenticación y gestión de usuarios	33
3.3. El acceso y los permisos sobre los objetos	35
3.3.1. El concepto de ACL (<i>aces control list</i>)	35
3.3.2. El nivel donde situar las ACL	36
3.3.3. Los roles preestablecidos o la personalización de permisos	38
3.4. Técnicas especiales de seguridad: la encriptación y la anonimización	41

4. Workflow y ciclo de vida de los documentos.....	42
4.1. Consideraciones generales	42
4.2. Los ciclos de vida de los documentos	42
4.3. Los flujos de trabajo y su relación con los documentos	44
4.4. El versionado	47
4.5. Los eventos o reglas	48
5. La gestión de los documentos físicos en los EDRMS.....	51
5.1. Consideraciones generales	51
5.2. Los objetos relacionados con los documentos físicos	51
5.2.1. Funcionalidades relacionadas con los objetos físicos	53
6. Aplicación de las políticas de retención y disposición.....	56
6.1. Consideraciones generales	56
6.2. Conceptos clave y su implementación	57
6.2.1. "Disposition schedules"	57
6.2.2. Asignación al File Plan	59
6.2.3. Los procesos de disposición	60
6.3. La preservación en los EDRMS	61
7. Búsquedas y visualizadores.....	62
7.1. Aspectos relacionados con la búsqueda	62
7.1.1. La búsqueda por atributos	62
7.1.2. Búsqueda por contenidos	63
7.1.3. Otras posibilidades	65
7.2. Visualizadores de documentos	66
8. Integración con otras aplicaciones.....	68
8.1. Integración predefinida	69
8.1.1. Interfaces de usuario	69
8.1.2. Aplicaciones ofimáticas	69
8.1.3. Aplicaciones de digitalización	70
8.1.4. Conectores	71
8.2. Integración de aplicaciones existentes	71
8.3. Creación de aplicaciones propias	72
Bibliografía.....	75

Introducción

Las herramientas de gestión documental EDRMS no son un tipo de software que se instale y que con una pequeña formación puedan comenzar a utilizarse. Por tanto, surge un tema importante ¿cómo abordar la implementación?

Lo normal es que, una vez seleccionado el proveedor de las herramientas informáticas, la implementación se plantee como un proyecto tecnológico.

En este módulo estudiaremos los aspectos más específicos de la implementación de un EDRMS, extrayendo las buenas prácticas aconsejadas por los expertos y centrándonos en las personalizaciones o parametrizaciones necesarias.

Concretamente, los temas que estudiamos son los siguientes: aspectos que debemos tener en cuenta en el plan de implementación; el modelo de información tomando como base los objetos documentales; estrategias de implementación del modelo de acceso a la plataforma tecnológica; *workflows* y ciclos de vida de los documentos; la gestión de los documentos físicos en un EDRMS; la aplicación de políticas de retención y disposición en los EDRMS; y la integración de los EDRMS con otras aplicaciones existentes.

Ved también

A este respecto, la metodología que se utiliza en la gestión de este tipo de proyectos la estudiaremos en el módulo 3, "Gestión de proyectos tecnológicos" de esta asignatura.

Objetivos

Los objetivos que debe alcanzar el estudiante con este módulo son los siguientes:

1. Conocer cuáles son los pasos a dar para implementar una herramienta después de la decisión de adquisición de una aplicación de gestión documental.
2. Conocer cuáles son las opciones de la tecnología para definir repositorios, utilizar los objetos predefinidos y su personalización.
3. Entender el tipo de parametrización o personalización a realizar, teniendo en cuenta las opciones de la tecnología.
4. Conocer los tipos o clases documentales.
5. Saber cómo efectuar la autenticación de usuarios.
6. Definir los perfiles de acceso y el permiso sobre los objetos.
7. Entender los conceptos de privacidad: encriptación y la anonimización.
8. Comprender cómo implementar las funcionalidades relacionadas con el ciclo de vida de los documentos: flujos de trabajo y su relación con los documentos, el versionado de estos, y los eventos o reglas que rigen el ciclo de vida de los mismos.
9. Conocer los aspectos más importantes de la gestión de documentos físicos en un EDRMS.
10. Identificar las decisiones de implementación que hay que tomar.
11. Entender cómo implementar las políticas de retención y disposición en los EDRMS.
12. Identificar y conocer los aspectos relacionados con la implementación de los aspectos relacionados con el uso de los documentos: búsquedas y visualizadores.
13. Reflexionar sobre las posibilidades y consecuencias de integración de los EDRMS con otras aplicaciones o desarrollos propios.

1. El plan de implementación

La implementación de un EDRMS puede constituir un proyecto aparte o constituir un subproyecto dentro de un plan o programa más amplio¹.

⁽¹⁾Ved el módulo 3, "Gestión de proyectos tecnológicos".

Lo que estudiaremos en este apartado es aplicable a cualquiera de los escenarios.

El plan de implementación definirá las fechas y las tareas a realizar antes de la puesta en producción de la herramienta. Para poder establecer un plan realista, es necesario tener una buena visión del proyecto. Esta visión toma su tiempo aunque se haya avanzado bastante en la fase de análisis previa a la selección de la herramienta.

Los expertos recomiendan no acelerar la fase de planificación del proyecto, en la que se debe entender el tipo de diseño que se requiere, para realmente cubrir las necesidades de la organización.

El plan de implementación debe abordarse de una forma sistematizada, lo que implica la asignación de responsabilidades y la elaboración de los documentos base de la implementación.

En el proyecto de implementación se deben considerar como mínimo los siguientes aspectos:

- las tareas a llevar a cabo,
- las responsabilidades en la implementación,
- la documentación que se generará.

1.1. ¿Qué actividades o tareas hay que llevar a cabo?

Entre las tareas que hay que planificar para la implementación se encuentran:

- Las tareas encaminadas a definir el diseño técnico detallado y la parametrización que se requiere.
- Las tareas necesarias para implementarlo.
- Las tareas de prueba y validación.
- Las acciones informativas y formativas necesarias antes de la puesta en producción.

1.1.1. Diseño técnico e implementación

Entre las primeras se encuentra el tiempo necesario para llegar a un acuerdo sobre qué tipo de diseño técnico se necesita. Es el tiempo que se requiere para encajar las funcionalidades de la herramienta seleccionada con las necesidades y requisitos de la organización. La organización debe decantarse por el mejor enfoque para asegurarse la aceptación del nuevo sistema, dentro de los límites presupuestarios y de tiempo para la implantación del proyecto.

Uno de los aspectos clave de esta fase es la decisión entre **personalización frente a programación**. Es muy habitual que en la implementación de este tipo de herramientas se plantee en algún momento que, para el cumplimiento de determinados requisitos, es mejor la programación a favor de la personalización y desarrollo de algunas aplicaciones fuera de las funcionalidades básicas de las herramientas o plataformas compradas. Existen argumentos tanto a favor como en contra de las dos opciones. Los primeros a favor de la programación se basan en el control y la no dependencia de la herramienta seleccionada; los segundos a favor de la personalización, en buscar el máximo rendimiento a las herramientas adquiridas y en la transición sin problemas a nuevas versiones y funcionalidades que el proveedor pueda ofrecer en el futuro.

En la mayor parte de los casos, la última decisión a este respecto corresponde a los departamentos de IT, responsables de la estrategia de tecnologías de la información y de su desarrollo y mantenimiento a lo largo del tiempo.

Repasaremos los escenarios más comunes donde se puede producir este planteamiento.

a) Aunque todas las herramientas EDRMS ofrecen una interfaz (o varias) para la interacción con el usuario final, es frecuente que se hagan planteamientos de construcción de una interfaz propia, que no solo incluya el diseño de logos y colores corporativos (esto se puede hacer normalmente sobre las interfaces de la herramienta), sino que permita una personalización mayor. Estas interfaces utilizan las funcionalidades del repositorio e interactúan con el mismo.

b) La creación de un bus de servicios web o *framework* que hace de enlace entre las aplicaciones de gestión y tramitación y el repositorio. Los EDRMS suelen ofrecer algunos de estos servicios e incluso conectores específicos para las tecnologías más conocidas, sin embargo, en muchos casos se plantea la creación de un bus de servicios específicos o *framework* que sea independiente del propio repositorio.

En los casos en que los departamentos TIC se decanten por una programación específica, se debe poner especial atención en cómo se va a implementar la lógica de la gestión de documentos. En muchos casos, en el afán por independizarse de la herramienta, los EDMRS están siendo utilizados como meros repositorios de ficheros, sin metadatos ni estructuración en el propio repositorio

(que se encuentran en las aplicaciones desarrolladas específicamente). Esta situación impide *a posteriori* la implementación de otras funcionalidades, como por ejemplo las de *records management* ofrecidas por las plataformas, llegando en algunos casos a dejar de tener sentido el uso de la tecnología EDRMS.

Tanto si la decisión es programación como personalización, una vez que el diseño está acordado, se debe planificar el tiempo necesario para implementarlo. Lo normal cuando se inicia un proyecto de este tipo es que se haya previsto una dedicación inicial del equipo técnico para la implementación y personalización basada en la experiencia de otros proyectos. Sin embargo, en muchos casos, una vez que se ha concretado en la fase previa qué es lo que realmente hay que hacer, puede ser necesario corregir el tiempo o los recursos previstos.

La implementación de un EDRMS en las administraciones públicas

En las administraciones públicas la implementación de un EDRMS se encuadra muchas veces en un presupuesto cerrado, que incluye las licencias de las herramientas y los recursos humanos (jornadas/horas) necesarios para la implementación. Cuando en la fase de diseño se descubre que las tareas de personalización eran mayores de las previstas para cumplir con las expectativas de los clientes, suele comenzar una ardua tarea de negociación; donde algunas de las funcionalidades deseadas quedan relegadas a una segunda fase del proyecto.

La mejor manera de evitar esta situación es no minimizar desde el principio la necesidad de apoyo en la fase de implementación, y haber trabajado algo por adelantado en la redacción de las propuestas. Cuanto mejor haya entendido el proveedor las necesidades, menos posibilidades existen de un mal entendimiento del trabajo a realizar.

Un enfoque muy utilizado por parte de los proveedores es la realización de un diseño inicial o “prototipo” que las distintas partes implicadas pueden probar y utilizar, al mismo tiempo que van entendiendo cómo pueden usar las funcionalidades que han ofrecido las distintas tecnologías.

1.1.2. Pruebas y validación

Una vez que se ha implementado el diseño acordado, es necesario establecer un periodo de pruebas antes del pase a producción.

En la implementación de EDMRS es muy importante que las pruebas de usuario las realicen los responsables de gestión documental además de los usuarios finales o responsables de los procesos a automatizar.

En estas pruebas debe comprobarse que todos los procesos documentales desde la creación de los documentos en los procesos de trabajo hasta la aplicación de las políticas de conservación funcionan como se habían diseñado.

En cualquier caso, antes de las pruebas de usuario final es necesario que el implementador o proveedor haya comprobado el correcto funcionamiento de todas las funcionalidades y su correcto funcionamiento.

Ejemplo

En el informe técnico ISO TR 22975 :2009, “Document management – Analysis, selection and implementation of electronic document management systems (EDMS)”, se presenta una tabla con recomendaciones sobre cómo realizar las pruebas en un proyecto de implementación:

Pruebas de integrador/proveedor	Pruebas técnicas de la organización	Pruebas de las unidades organizativas
Pruebas de unidades/módulos: Según cada unidad o módulo es implementado, debe realizarse una prueba completa de todas las funcionalidades.	Las actualizaciones de cada módulo se prueban para supervisar el progreso de la implantación.	Se prueban todos los aspectos y consideraciones del diseño para asegurarse de que el sistema cumple con las necesidades y requerimientos del usuario final.
Pruebas de sistema: Según el sistema es configurado en el entorno de prueba, el integrador o proveedor debería probar todos los aspectos del mismo.	Las actualizaciones del sistema se prueban para supervisar el progreso del proyecto.	Las actualizaciones en el progreso del proyecto mientras se continúan preparando para la implementación y se ejecutan los aspectos relacionados con el cambio en la organización.
	Prueba completa del sistema: Después de que el integrador/proveedor haya completado sus pruebas, el equipo técnico de la organización debería realizar una prueba detallada para asegurar la conformidad con las expectativas de la organización y con los requisitos funcionales y técnicos.	
		Pruebas de usuario final: Después de que el equipo técnico de la organización haya completado con éxito sus pruebas, una selección de usuarios finales debería probar el sistema desde la perspectiva de la organización para asegurarse de que todas las funciones funcionan como se había anticipado.

Fuente: TR ISO 22975:2009. Traducción propia.

Actividad

Estudiando la tabla anterior, ¿en qué tipo de las pruebas que nos proponen estarían encuadrados los especialistas de gestión documental?

Durante el periodo de pruebas deben seguirse unos criterios generales, que se aplican a cualquier implantación:

a) Durante las pruebas de aceptación, la organización mantiene un registro de eventos e identifica cualquier deficiencia de software o hardware.

b) No se debe producir ninguna modificación del hardware o del software sin el consentimiento y aprobación del jefe de proyecto. La organización debe conceder un tiempo limitado, pero suficiente para resolver los problemas surgidos durante las pruebas.

c) La suspensión de las pruebas de aceptación solo debería ocurrir por mutuo acuerdo o si la organización decide que la solución no está lista para las pruebas. Si esto ocurre, debe planificarse cuándo se podrán reiniciar, una vez que se hayan subsanado las deficiencias.

d) Al final de la prueba de aceptación el jefe de proyecto revisa la lista de deficiencias, si existiese alguna, y decide:

- Aceptar el sistema asumiendo una fecha de corrección para las posibles deficiencias.
- Rechazarlo y fijar otras fechas para las nuevas pruebas.

1.1.3. Acciones de gestión del cambio

La implantación de un EDRMS debe ir siempre acompañada de acciones de gestión del cambio. El paso de los documentos en papel a los documentos electrónicos se trata no sólo de un cambio en los procesos de trabajo, sino también en las actitudes de las personas. Por esto, es imprescindible que el plan de implementación incluya las distintas acciones de gestión del cambio planificadas.

Uno de los aspectos más importantes a tener en cuenta es cómo se plantea el escenario de transición en el que las personas convivirán con documentos en papel y documentos electrónicos durante bastante tiempo.

En cualquier caso la gestión del cambio se debe basar en dos tipos de acciones:

- Las acciones de formación y comunicación.
- La redacción de documentación de apoyo, en la que se describen los procedimientos con los documentos electrónicos.

Las acciones de formación y comunicación que deben acompañar a cualquier implantación de un EDRMS deberían establecerse de la siguiente forma:

1) **Identificación de los destinatarios.** Se deben identificar los distintos roles de las personas en virtud de su relación con la gestión de los documentos electrónicos. Los roles básicos que se pueden identificar en todas las organizaciones son:

- Las personas de las unidades organizativas, que interactúan y crean documentos electrónicos.

Lectura complementaria

Para tener información complementaria sobre la gestión del cambio, podéis ver:

J. A. Alonso Martínez; M. García Alsina (2013). *Análisis y diseño de un modelo de gestión documental*. Barcelona: FUOC.

- Los jefes de las unidades administrativas, que deben entender el nuevo escenario para poder controlar y motivar a los equipos correspondientes.
- El personal de unidades organizativas transversales, que deben apoyar y a veces intervenir en diferentes procesos de trabajo.

2) Identificación del tipo de acción y mensajes a transmitir. Las acciones pueden ser:

- **Formativas generales.** Tienen como objetivo el conocimiento de las distintas implicaciones de la gestión de los documentos electrónicos y de la utilización de los EDRMS.
- **Formativas específicas.** Tienen como objetivo el aprendizaje de la nueva herramienta y cómo se utiliza.
- **Informativas.** Tienen como objetivo informar cuáles son los objetivos perseguidos, cuáles son los planes para alcanzarlos y cuál es la visión general de la organización.
- **Divulgativas.** Tienen como objetivo informar sobre los avances en la implantación del proyecto o sus fases.

3) Identificación de las distintas herramientas que se utilizarán para llevar a cabo las acciones. En cada organización pueden existir muchas diferentes. A modo de ejemplo, podemos mencionar:

- Cursos formales, incluidos por ejemplo en un plan de formación.
- Sesiones formativas asociadas a la implantación de una herramienta.
- Charlas o presentaciones.
- Autoformación *on-line*.
- Sección en la intranet.
- Material divulgativo.

Las acciones planificadas deben estar adaptadas a las circunstancias, medios y posibilidades de cada organización, pero existen una serie de recomendaciones derivadas de las buenas prácticas que deberían tenerse en cuenta:

- Los programas formativos que habitualmente se asocian a la implantación de la nueva herramienta informática no deben concentrarse solo en el uso de la herramienta y sus posibilidades, deben aprovecharse para encuadrar los principios y objetivos de la organización con respecto a la gestión de los documentos electrónicos y los contenidos, y concienciar sobre la importancia de la actuación de cada una de las personas en consonancia con las políticas y procedimientos establecidos.
- Es muy importante que los principios básicos de uso de este tipo de herramientas se incluyan en los protocolos de acogida de las personas que se integran por primera vez en las organizaciones.

1.2. Definición de responsabilidades

En el plan de implementación deben quedar claramente definidas las responsabilidades sobre cada una de las tareas.

En los proyectos de implementación de un EDRMS además de los usuarios finales, debemos procurar siempre un equipo multidisciplinar. En el caso de los usuarios finales, es recomendable segmentarlos al menos como se han definido en el apartado anterior (identificación de los destinatarios).

Como mínimo, en el equipo multidisciplinar deben estar los responsables de gestión documental y los responsables informáticos. El jefe de proyecto puede recaer en cualquiera de los dos perfiles mencionados o incluso en algunos casos se establece una jefatura de proyecto compartida.

En otros casos, los proyectos se enfocan como meramente tecnológicos, considerando a los responsables de gestión documental como los usuarios finales. Esta aproximación no da tan buenos resultados, pero a veces es la única posible cuando las responsabilidades de gestión documental no están claramente asignadas e identificadas en la organización.

Entre las responsabilidades a asignar están cada una de las tareas descritas en este apartado: el diseño técnico, las pruebas y la validación, las acciones de gestión del cambio y la generación de la documentación.

1.3. Documentación

La documentación que se debe generar en la implementación de un EDRMS es al menos de dos tipos:

- La documentación del proyecto de implementación, que recoge las decisiones tomadas y los distintos pasos que se han ido dando.
- La documentación destinada al usuario final, para ayudarle en la utilización de la herramienta y en adaptarse a la nueva manera de gestionar los documentos.

Entre los primeros nos podemos encontrar, según las metodologías, distintas denominaciones de los documentos. Las denominaciones más extendidas y utilizadas son:

- Análisis de requisitos
- Análisis funcional
- Diseño técnico
- Plan de pruebas

Ved también

Como se estudiará en el módulo 3, "Gestión de proyectos", estos son documentos o entregables propios de cualquier proyecto tecnológico.

La implantación de un EDRMS, aunque sea basado en una herramienta de mercado, requiere de una fase de configuración importante, por lo que muchas veces los entregables se asimilan a un proyecto de desarrollo tecnológico.

En muchos casos, se entiende que el análisis de requisitos ya se ha realizado cuando se ha seleccionado la herramienta, y por lo tanto, no es un entregable propio del plan de implementación.

No obstante, sí es totalmente recomendable la redacción y validación de un **análisis funcional y/o diseño técnico**. Dependiendo de las decisiones tomadas con respecto a las necesidades de programación e integración con otras herramientas, serán necesarios ambos, o se podrá recoger toda la información en un solo documento.

Fundamentalmente, este/os documento/s debe/n contener todas las decisiones tomadas con respecto a la configuración. Analizaremos todos los aspectos sobre los que hay que tomar decisiones en los próximos apartados.

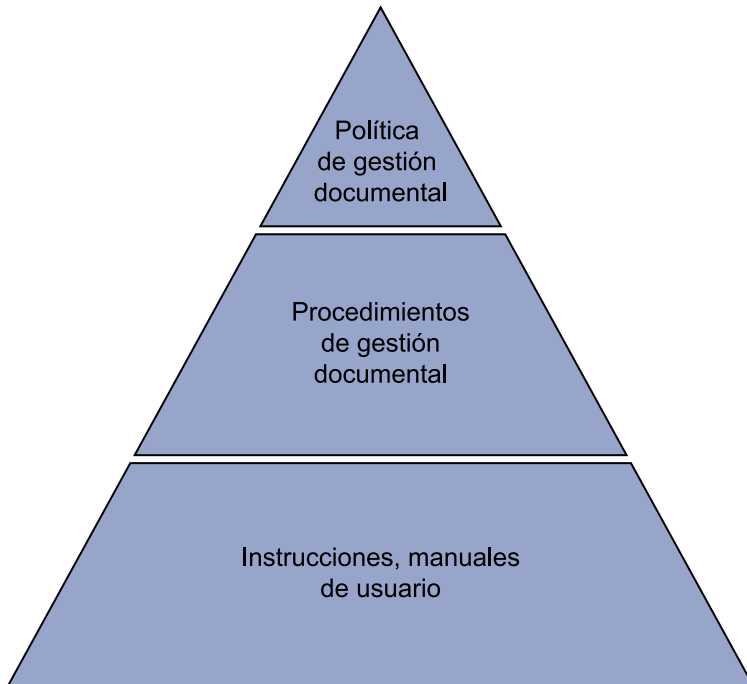
Es importante con respecto a estos documentos que reflejen exactamente lo que se ha implementado, manteniéndose actualizados, incluyendo cualquier tipo de modificación que se haya tenido que introducir durante las fases de prueba y validación.

Es especialmente interesante que recojan las alternativas descartadas y las razones para haberlas descartado, así como los puntos o funcionalidades deseables que se posponen para siguientes versiones o fases.

Entre la documentación destinada al usuario final, deberemos adaptarla a las necesidades de la organización.

Normalmente, las herramientas proveen de manuales de usuario final, pero estos no son útiles cuando se ha producido una personalización y configuración importante. En las organizaciones acostumbradas a documentar sus procesos, la documentación de usuario debería adaptarse a la terminología y estructura utilizada. En la figura 1 encontramos el ejemplo más sencillo de una documentación en soporte de la gestión documental.

Figura 1. Pirámide de documentación



Fuente: Elaboración propia

Muchas veces, el proceso de implementación de una herramienta EDRMS es el acicate para la realización de este tipo de documentación.

1.4. Despliegue

Aunque la selección de la herramienta y el análisis y configuración conviene afrontarla con la visión global de toda la organización, el despliegue de la solución conviene hacerlo por etapas o fases.

En muchos proyectos se planta una fase “piloto” en la que la solución se circunscribe a unas personas y procesos determinados, con la intención de valorar los resultados y plantear correcciones o mejoras antes del salto a la siguiente fase.

Para planificar cada una de las fases del despliegue de una herramienta EDRMS, es necesario lo siguiente:

- Comprender cuáles son las acciones de gestión del cambio que deben acompañar a cada fase del despliegue.
- Enmarcar el despliegue del proyecto de implantación del EDRMS con las demás acciones y actividades que se estén llevando a cabo.

Es habitual que el despliegue se haga por procesos o procedimientos automatizados; pero también pueden establecerse por unidades organizativas. Lo importante es tener en cuenta que no se puede realizar la implantación y el comienzo de la utilización del nuevo entorno sin que se cumplan las siguientes condiciones:

- Que todas las personas y perfiles involucrados hayan recibido la información/capacitación necesaria para trabajar con la nueva herramienta.
- Que estén a su disposición la documentación de soporte que explica cómo se debe proceder en el nuevo entorno.

2. Modelo de información: los objetos documentales

En este apartado estudiamos estrategias para entender las opciones que la solución tecnológica nos ofrece: repositorios, la diferenciación entre *document* y *record*, la utilización de objetos predefinidos, y los tipos o clases documentales que se deben tener en cuenta para la implementación de los metadatos.

2.1. Entendiendo las opciones de la tecnología

En el corazón de cualquier solución EDRMS nos vamos a encontrar con los servicios de repositorio o librería. Todas ellas parten de unos principios comunes, aunque utilicen terminologías particulares, con las que, sin duda, deberemos familiarizarnos y entender antes de proceder a su configuración o personalización.

Entendiendo el repositorio como el lugar donde se almacenan los contenidos y documentos electrónicos, los EDRMS nos proveen de un modelo de información para organizar ese repositorio.

En la ingeniería de software, un modelo de información es una representación de conceptos y sus relaciones, así como las reglas, restricciones y operaciones especificadas para un dominio concreto.

Un modelo de información es imprescindible cuando se diseña un software específico partiendo de cero. En estos casos, el modelo de información puede llegar a ser complejo y tener varios niveles, normalmente se utilizarán lenguajes de modelaje de la información estándar para llevar a cabo esta tarea, que se realiza por parte de expertos en el desarrollo de software, que traducen los requisitos de los usuarios.

En la práctica, la puesta en funcionamiento de un EDRMS parte de un modelo de información preestablecido al que nos tendremos que adaptar, pero con un margen importante de personalización, que es lo que estudiaremos en este apartado.

El primer paso que hay que dar es, por lo tanto, familiarizarse con la herramienta y su terminología. Los proveedores de los EDRMS facilitan manuales de usuario y formación específica en la misma. Normalmente, ambos están divididos en niveles de administradores y usuarios. El entendimiento del modelo de información y sus conceptos está en un terreno intermedio, que bien puede ser ofrecido como el de usuario avanzado, o como de administradores funcionales.

Utilizando algunos EDRMS de ejemplo, vamos a estudiar cuáles son los conceptos comunes a todos ellos.

En general los EDRMS siguen el diseño **orientado a objetos**. Esto quiere decir que cualquier elemento que se incluye en el sistema (tanto un documento con sus metadatos, como la definición de un determinado *workflow*, o la definición de un determinado tipo de documento) existe en el repositorio como un objeto. De esta forma, cada objeto contiene encapsulados los detalles específicos que conforman tanto su identidad como su comportamiento. El conjunto de los objetos y cómo se relacionan entre sí es lo que permite ofrecer unas determinadas funcionalidades.

Las características de los objetos se basan en:

- los metadatos que definen su identidad, lo que permite agruparlos por tipos;
- las relaciones entre los objetos (por ejemplo una relación jerárquica permite heredar las características del padre);
- los patrones de acceso a los objetos;
- el comportamiento de los mismos.

Los documentos o contenidos electrónicos serán, por lo tanto, objetos dentro del sistema, que estarán compuestos por un fichero o archivo que tiene el contenido y una serie de metadatos que habitualmente se almacenan en una base de datos asociada al EDRMS.

Nota

Los EDRMS se soportan habitualmente en una base de datos relacional, donde se almacenan los metadatos que forman parte de los objetos. Actualmente, casi todos los productos del mercado son compatibles con las bases de datos más habituales del mercado, como SQL u ORACLE. Cuando se instalan los sistemas, estos se encargan de crear las estructuras de datos necesarias en las bases de datos asociadas, de manera que normalmente no hay que intervenir sobre ellas.

Todos los EDRMS nos van a proporcionar unos determinados tipos de objetos por defecto o base, que han sido predefinidos para ofrecer las funcionalidades de gestión documental tal y como las entienden las distintas aplicaciones.

Los EDRMS nos ofrecen los tipos de objetos, que normalmente pueden ser personalizados, para construir nuestras propias soluciones. Además, estas plataformas son lo suficientemente abiertas para permitir el desarrollo de nuevos objetos o la integración con otras aplicaciones. Solo entendiendo cómo funciona la tecnología que vamos a implementar conseguiremos el máximo rendimiento de la misma, utilizando su máxima potencialidad sin tener que “reinventar” las funcionalidades.

No hay ninguna normalización en el lenguaje utilizado por unas herramientas u otras, aunque los conceptos sean similares o muy próximos. Términos que para una determinada herramienta tienen un sentido, pueden tener otro completamente distinto para su competidor. Antes de implementar un EDRMS, nunca se debe dar por supuesto que se entiende la terminología utilizada.

Toda esta definición previa, necesaria para la implementación del EDRMS, se debe documentar, tal como hemos visto en el apartado anterior² en un análisis funcional y/o diseño técnico.

⁽²⁾Ved el apartado 2.3, "Documentación".

Esta definición se realiza la primera vez que se implementa la herramienta, pero debe ser mantenida a lo largo del tiempo, según se efectúa el despliegue de la misma. Por eso está cada vez más claro que esta definición forma parte de lo que se ha llamado *information governance* o modelo operativo para el uso de la herramienta.

SharePoint Governance

Sobre una de las herramientas más extendidas, Microsoft Sharepoint, se ha desarrollado una gran literatura sobre SharePoint Governance. Podéis leer en la web de AIIM lo que se consideran los puntos clave de esta gobernanza. <http://www.aiim.org/Community/Wiki/view/Quick-Guide-to-SharePoint-Governance>

2.2. La definición de los repositorios o espacios

La primera tarea que hay que realizar es la definición de la arquitectura del repositorio. Esta arquitectura tiene una parte física, que dependerá de la infraestructura informática de servidores y redes de la organización, y otra lógica, que es la que vamos a estudiar en este apartado.

La idea central sería cómo se van a organizar los contenidos y documentos en distintos niveles. Aunque con pequeños matices, todas las herramientas permiten establecer un primer nivel de organización que indica una cierta agrupación de documentos.

En Documentum correspondería a las Docbases, en Filenet a los Document Object Store, en Sharepoint a los Sites o Sites Collection y en Opent Text a los Workspaces. La lógica responde a que todos los documentos que se integren en un mismo nivel van a compartir una lógica de funcionamiento y diseño de objetos documentales.

En muchos manuales de implementación de este tipo de herramientas nos podemos encontrar una identificación de este primer nivel con los distintos departamentos o áreas de una organización. Por ejemplo: administración, legal, operación, tecnología, etc. Esta es una posibilidad, pero no necesariamente la única.

















En algunas herramientas el volumen esperado de documentos puede ser un criterio para establecer distintas instancias de este tipo. En este caso hay que seguir las indicaciones del fabricante o implementador.

En algunas organizaciones pequeñas es incluso posible que no sea necesario este primer nivel de división, que, sin embargo, puede ser muy útil en grandes organizaciones, con lugares geográficos distantes y oficinas regionales con cierta autonomía, o en una corporación con distintas compañías o negocios que quieren compartir la plataforma de gestión de contenidos.

A partir de esta primera división se debe estudiar la lógica con la que vamos a organizar el repositorio, contando con los tipos de objetos que la herramienta nos ofrezca.

A continuación se presentan en las figuras de la 2 a la 4 algunos ejemplos de los objetos que pueden utilizarse en las distintas herramientas.

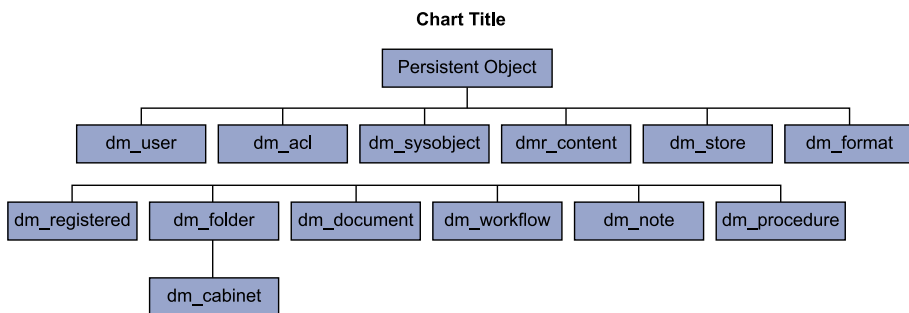
Figura 2. Extracto de los tipos de items. Open Text Content Server

Icon	Name	Description
	Document	A Document of undetermined type. By default, Content Server recognizes many file types, including  Microsoft® Word document files,  Microsoft® Excel spreadsheet files, and  HTML document files.
	Text Document	A document file in plain text format.
	News	A News item added to a Channel that can be broadcast in a News Player. News is made up of a headline and a story.
	Topic	The beginning of a thread in a Discussion.
	Reply	The continuation of a thread in a Discussion.
	Task	A work assignment issued to a user or group.
	Milestone	A marker used to associate a Task with a particular stage of a Project or date.
	Workflow Map	A graphical depiction of a structured work process.
	Poll	An item that enables users to vote on specific issues.
	LiveReport	A direct query run against the Content Server database that reports statistics and other information about the database and the items in it.
	Query	A stored set of search criteria that can be re-executed.
	Snapshot	A stored set of Search Results. Displaying the Snapshot does not re-execute the Search that produced it.
	Version	A previous version of a Document, Workflow Map, or other versionable item type that is maintained by Content Server.

Icon	Name	Description
	Appearance	A special container that stores Documents and special Text Documents containing HTML code, which are used to apply customizations to the Content Server user interface.
	Folder	A container that stores other items. Folders can contain almost all types of items. A Folder inside another Folder is known as a subfolder.
	Compound Document	A container that stores multiple Documents, such as chapters of a book, in sequential order. Compound Documents can contain only Documents, Shortcuts, Generations, and other Compound Documents.
	Channel	A container for storing and publishing News items.
	Project	A work environment for teams who are collaborating on a common goal.
	Discussion	A container that holds collections of Topics and Replies by which users exchange information about a common subject.
	Task List	A container that stores and organizes Tasks.
	Task Group	A container within a Task List used to organize large groups of related Tasks.
	Category	A container for storing and organizing attributes.
	Custom View	A special container for storing HTML code that enables you to customize the appearance and behavior of Content Server.

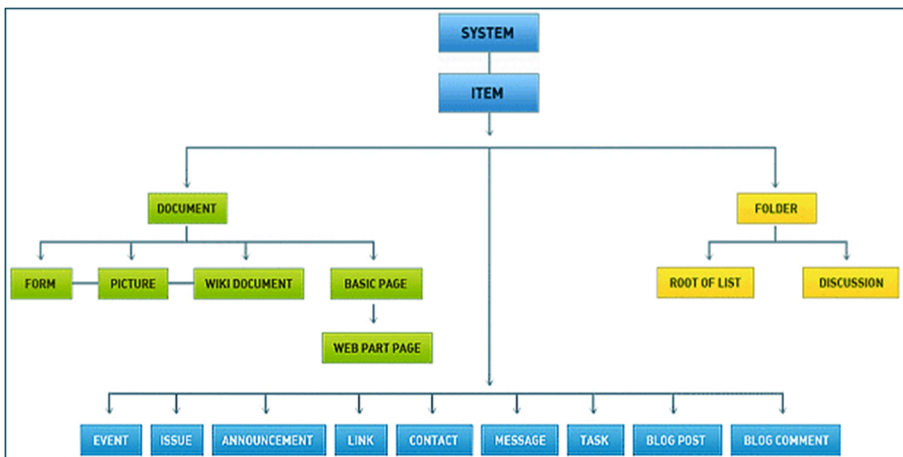
Fuente: Open Text Content Server: User Online Help

Figura 3. Árbol jerárquico de tipos de objetos en Documentum



Fuente: Elaboración propia

Figura 4. Microsoft Sharepoint content types



Fuente: <http://www.offshoredotnetdevelopment.com/share-point-services.shtml>

Como podemos desprender de los ejemplos de las tres figuras, cada una de las herramientas utiliza su propia lógica y terminología para establecer sus objetos, sin embargo, en todas ellas podemos encontrar dos objetos básicos que casi siempre se llaman igual:

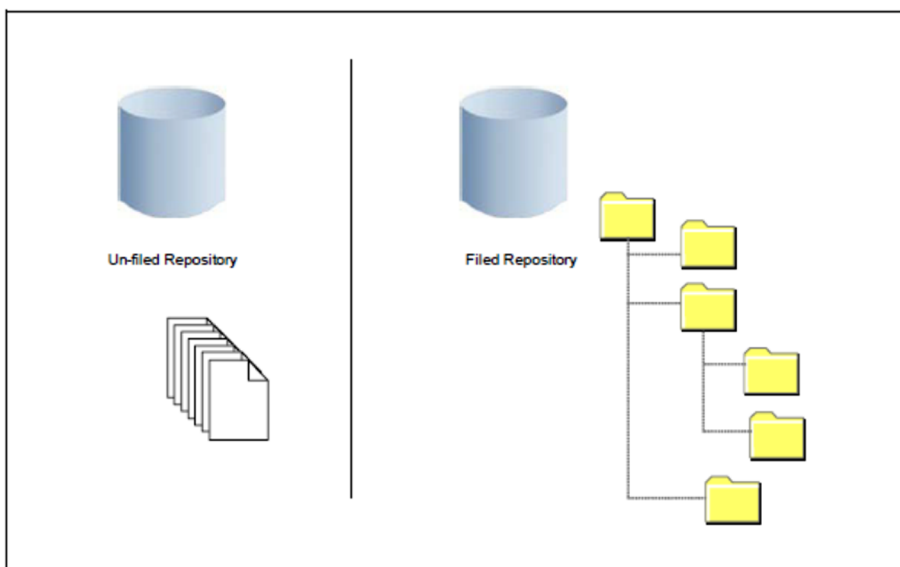
- **Document**, que representaría el tipo de contenido más común compuesto por un fichero o archivo y sus metadatos.

- **Folder**, que representa a un objeto contenedor de otros objetos.

Estos son los dos elementos básicos sobre los que se asienta la organización de los repositorios de documentos electrónicos. En algunos casos, estos mismos objetos pueden expandirse en algunos más, por ejemplo en Sharepoint derivados de Document, tenemos Picture o wiki document, mientras que en Documentum tenemos un objeto contenedor derivado de *folder* que se llama Cabinet, y cuyo icono es el típico archivador que contiene carpetas.

La primera decisión a tomar es si se va a implementar un repositorio organizado con las posibilidades que nos ofrecen los objetos contenedores tipo *folder* o no. La diferencia se representa en la figura 5.

Figura 5. Diferentes aproximaciones de cómo configurar un repositorio según Filenet



Fuente: IBM FileNet Content Manager Implementation Best Practices and Recommendations

Existen implementaciones de los ECM en los dos tipos de aproximaciones, pero la primera es más frecuente cuando no se permite un acceso directo a los repositorios y toda la lógica de organización reside en aplicaciones externas.

La utilización de objetos contenedores nos permitirá aplicar relaciones, reglas y metadatos a todos los objetos agrupados en un mismo contenedor, lo que a la larga suele facilitar enormemente la gestión.

Desde la gestión documental se ha visto cómo esta estructuración de los repositorios puede ligarse con el proceso documental de la clasificación. En esta aproximación se intentan utilizar los cuadros de clasificación documental para estructurar los repositorios desde un inicio. Esta opción está siendo ampliamente utilizada en algunas implementaciones.

Si se decide no optar por clasificaciones ya existentes, como pueden ser el cuadro de clasificación documental, habrá que construir otra organización específica para organizar el repositorio.

2.3. La diferencia entre *document* y *records*; el paso de *document* a *record*

En la mayor parte de las herramientas que existen en el mercado las funcionalidades de *records management* se comercializan y, por lo tanto, se implementan como módulos separados.

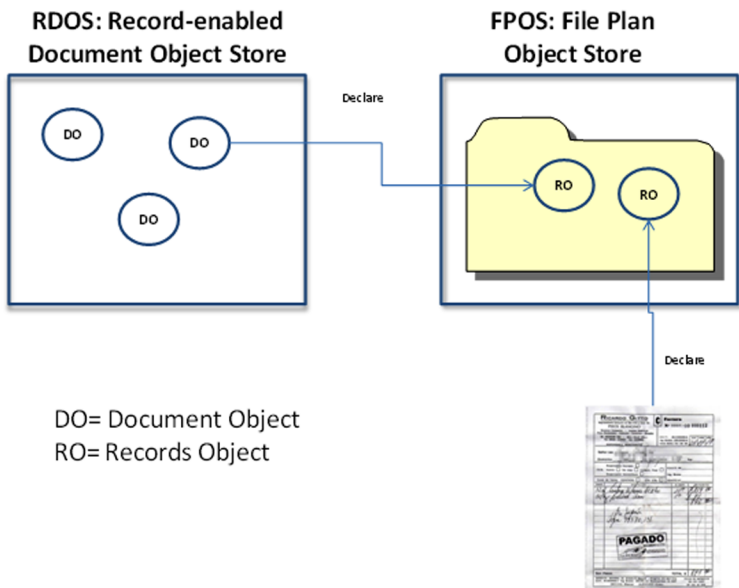
En la lógica que comparten todos los EDRMS existe un momento en que un documento, es decir, una instancia del tipo de objeto *document* puede ser marcado o convertido en *record* (declarado como *record* en una traducción directa del término utilizado en inglés). En algunos será el cambio de un metadato en un documento, en otros será la creación de un nuevo objeto y en otros será el cambio de repositorio. En todas las herramientas existentes actualmente en el mercado se nos ofrecen funcionalidades para automatizar este proceso de “declaración de record”, si bien se sigue ofreciendo la posibilidad de declaración “manual”.

2.3.1. Cambio de tipo de objeto

Cuando el paso de *document* a *record* supone un cambio de tipo de objeto, el objeto tipo *record* tiene una información y un comportamiento diferente al tipo de objeto *document*. Este objeto está también definido por defecto. Además, cuando se produce este cambio, se deben contemplar otros aspectos, como la creación de metadatos o la clasificación de los documentos, o la relación entre los objetos, como vemos a continuación.

En algunos casos, además supone que ese objeto *record* se guarda en otro repositorio específico para el almacenamiento de *records*. En la figura 6 se esquematiza el funcionamiento de Filenet P8 en este sentido. Los documentos electrónicos al declararse como *record* cambian de tipo de objeto y de repositorio. Los documentos físicos, que estudiaremos en un apartado más adelante, se incorporan directamente como *records objects*.

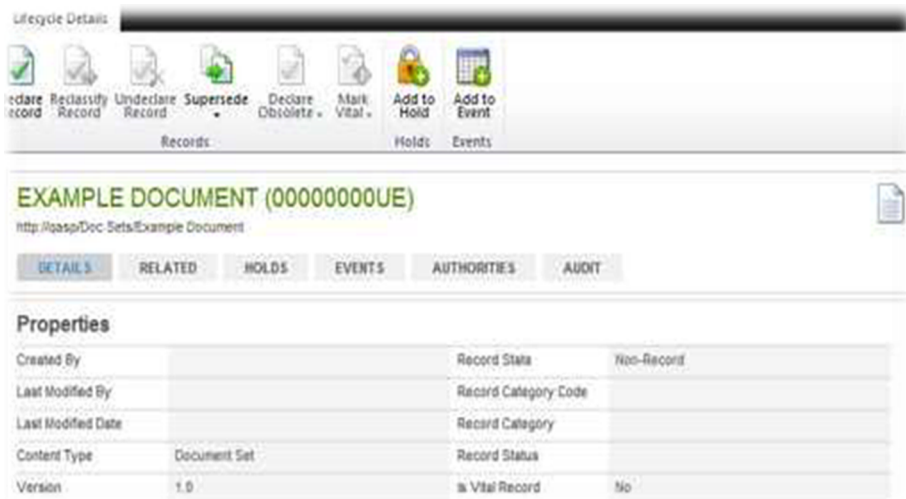
Figura 6. Records Objects en Filenet P8



Fuente: Elaboración propia

Lo que todas las herramientas tienen en común es que una vez los documentos son *records*, se incluyen nuevos metadatos propios de los *records*. En la figura 7 se muestra una captura de pantalla de la herramienta Collabware (*records management* para Sharepoint) con los atributos propios de *record*.

Figura 7. Pantallazo de las propiedades de un Record en Sharepoint usando Collabware



2.3.2. File Plan o clasificación

Otra parte importante del módulo de *records management* que es común en todas las herramientas es la gestión del “File Plan” o clasificación. Normalmente está implementado sobre un tipo de objeto contenedor específico, derivado del objeto *folder*. En Filenet, por ejemplo, se llaman *records class*, y en Collabware Sharepoint *records categories*. Dado que no existe una traducción única

para este concepto que utilicen todas las herramientas, en este material vamos a seguir utilizando la denominación en inglés *file plan*, ya que en este idioma la terminología es común en todas las herramientas del mercado.

Entendiendo cómo funciona la herramienta, debemos tomar las decisiones necesarias de cómo utilizar o no el módulo de *records management*.

2.3.3. Relación entre los objetos

Cuando se decide implementar el módulo de *records management* tendremos que establecer la relación y comportamiento de los distintos objetos y cómo se producirá la declaración de *records*. Este módulo nos permitirá automatizar las políticas de disposición como veremos en su apartado correspondiente.

Los puntos a tener en cuenta son:

1) La alineación o no de la estructura del repositorio base, con la estructura del *file plan* del módulo *records management*. Las posibilidades son las siguientes:

- Si se decide tener un repositorio base no estructurado, la declaración de *records* supondrá la clasificación de los mismos en el *file plan*. Para poder automatizar esta clasificación deberá recurrirse a otros parámetros, como los tipos de objetos o el valor de un metadato. Establecer esta correspondencia es complicado si no se ha previsto desde el inicio.
- Mantener dos clasificaciones diferentes: una para el repositorio base y el *file plan* del módulo de *records*. Esta opción se sustenta en la necesidad de una estructura más apegada al negocio durante la realización de la actividad o el proceso. La dificultad estriba en el mantenimiento de dos estructuras clasificatorias que hay que gobernar al mismo tiempo.
- Sincronizar la estructuración del repositorio inicial con el *file plan*. La ventaja es una declaración automática sin problemas. La desventaja puede ser la necesidad de flexibilizar el *file plan* para adaptarlo a las necesidades del negocio.

2) La definición de las reglas de automatización de la declaración de *records*. Normalmente los propios módulos de *records management* nos ofrecerán esta funcionalidad, que debemos aprovechar al máximo.

3) El entendimiento de las consecuencias de la declaración de *record* en las condiciones de acceso a los documentos. La funcionalidad principal que las herramientas proveen es que una vez que se ha declarado un *record* los permisos de modificación del documento desaparecen; pero lo normal es que todos

los permisos de acceso sean sustituidos por los que corresponden a la categoría o clase del *file plan* donde ha sido clasificado. Lo estudiaremos más en profundidad en el apartado de acceso.

4) Cómo se van a tratar los metadatos o información complementaria que se haya asignado a los distintos tipos de objeto.

En cualquier caso hay que tener presente que normalmente la “declaración de *records*” solo se aplica sobre el objeto de base *document*.

2.4. La utilización de objetos predefinidos y su personalización

Aunque es posible usar solamente los objetos que vienen definidos en las herramientas EDRMS, lo habitual es una configuración personalizada. Esta personalización puede tener dos vertientes complementarias:

- La definición de tipos de objetos que se derivan de los existentes por defecto en las herramientas, lo que está normalmente relacionado con la implementación de los metadatos que veremos en el apartado siguiente.
- La definición de nuevos objetos personalizados, que se definen con unas características y comportamientos propios.

Las herramientas o plataformas tecnológicas nos van a proveer de herramientas que posibilitan las dos acciones, si bien lo más normal es que la personalización de objetos preexistentes pueda ser realizada por un administrador funcional que haya recibido la formación adecuada en la herramienta; para la creación de nuevos tipos de objetos se requerirán conocimientos técnicos específicos.

La decisión de cómo realizar esta personalización debe basarse en un equilibrio entre los siguientes criterios:

- La adecuación a los requisitos de la organización.
- La capacidad de la organización para llevarlos a cabo y mantenerlos.
- Las consecuencias que la personalización puede tener en la actualización posterior de las herramientas.

Ejemplo

En la implementación de herramientas EDRMS en las administraciones públicas españolas, y al amparo del esquema nacional de interoperabilidad, se ha utilizado en más de una ocasión el siguiente modelo de personalización de los tipos de objetos:

- Tipo de objeto documento. Se utiliza el tipo de objeto *document* de base, pudiéndose establecer varios tipos diferentes a partir del mismo.
- Tipo de objeto expediente. Se utiliza el tipo de objeto *folder* para representar el concepto de expediente electrónico.
- Tipo de objeto firma. Se crea un nuevo tipo de objeto para las firmas asociadas a los documentos. Cuando la firma proviene de un certificado digital *detached* (separada del objeto que firma), estará compuesta por un fichero digital al que se asocian metadatos. Sin embargo, cuando se opta por una firma *attached* (que se incluye en el objeto firmado) mediante certificado, esta estaría compuesta solo por los metadatos de la firma. Una firma solo puede ser de un documento, pero un documento puede tener más de una firma.

Cuando se implementa con módulo RM con este modelo de base, los que se pueden declarar como *records* son solo los del primer tipo de objeto. En este escenario hay que decidir cómo aplicar el modelo en relación con los otros objetos, como pueden ser el tipo de objeto *folder* utilizado como expediente, o el nuevo tipo de objeto firma dependiente de los documentos.

Además, la personalización de los tipos de objetos implica la definición de las relaciones y comportamiento de los objetos entre sí. Uno de los ejemplos más claros se produce en la relación entre los tipos de objeto *document* y los tipos de objeto *folder*. En algunas implementaciones se ha llamado a esto las “reglas de contención”.

El caso más claro es la decisión de si un documento puede estar contenido en más de una carpeta o *folder*. La tecnología nos va a permitir todas las posibilidades, pero en el proceso de personalización deberemos elegir la que más se ajuste a los requisitos de la organización.

En las implementaciones en las administraciones públicas se ha discutido mucho sobre estas opciones para posibilitar el tratamiento de los documentos que pueden formar parte de más de un expediente. No todas las implementaciones han concluido lo mismo y podemos encontrar decisiones de dos tipos:

- Determinar que, a imitación del mundo en papel, cuando esto sucede se debe hacer una copia del documento electrónico.
- Determinar que el mismo objeto que representa el documento debe utilizarse para todos los expedientes en los que se necesite.

En otros casos, son las propias herramientas las que nos proveen de tipos de objeto “referencia” o “alias”, que se utilizan precisamente para remitir a otros objetos sin necesidad de hacer copias de los mismos.

2.5. Los tipos o clase documentales: cómo implementar los metadatos

Una de las partes más importantes de la definición del modelo de información en un proyecto de implantación de un EDRMS es la inclusión de los metadatos en el mismo.

A este punto deberíamos llegar con una idea clara, bien del esquema de metadatos que vamos a utilizar, bien con una definición clara de cómo definir los de metadatos.

Si no tenemos claro el modelo conceptual de metadatos a implantar y nos guiamos en la implementación por los deseos puntuales de los usuarios finales, crearemos una infraestructura de metadatos no sistemática y difícil de mantener en el tiempo.

Ved también

Esto se estudia en profundidad en otras asignaturas, como:

Procesos y herramientas de gestión documental (C. Bustelo Rueda; M. Juncà Campdepadrós, 2013).

Análisis y diseño documental (J. A. Alonso Martínez; M. García Alsina, 2013).

2.5.1. Pilares básicos

Para implementar los metadatos de forma eficaz, es necesario lo siguiente:

- tener claro el esquema o modelo de metadatos,
- conocer la herramienta en profundidad y
- conocer la arquitectura general de los sistemas de información de la organización.

2.5.2. Construcción del perfil de aplicación

Basándonos en estos tres pilares podremos construir el perfil de aplicación de metadatos, identificar en qué sistemas de información se encuentra la información, y buscar la forma de implementar los elementos de metadatos.

Un **perfil de aplicación** es el conjunto de elementos de metadatos y reglas de uso que se definen para una aplicación particular. Un perfil de aplicación es más fácil de hacer si se parte de un esquema de metadatos existente. Si cuando vamos a implementar el EDRMS no existe un esquema de metadatos en el que basarse, debemos construir directamente el perfil de aplicación, para lo cual es necesario al menos partir de un modelo predefinido. En este caso es muy útil utilizar la ISO 23081-2.

En los EDRMS muchos de los metadatos de gestión de documentos ya han sido implementados por defecto, así que lo primero que tendríamos que estudiar a fondo es cuáles son estos metadatos y mapearlos con los elementos del esquema de metadatos escogido.

En segundo lugar, tendremos que identificar en qué sistemas de información se encuentra la información que se corresponde con otros elementos del esquema de metadatos y decidir, conjuntamente con los responsables de esos sistemas de información, si se pueden establecer vínculos persistentes con el EDRMS a lo largo del tiempo; o si bien es preferible capturar la información y gestionarla dentro del propio EDRMS.

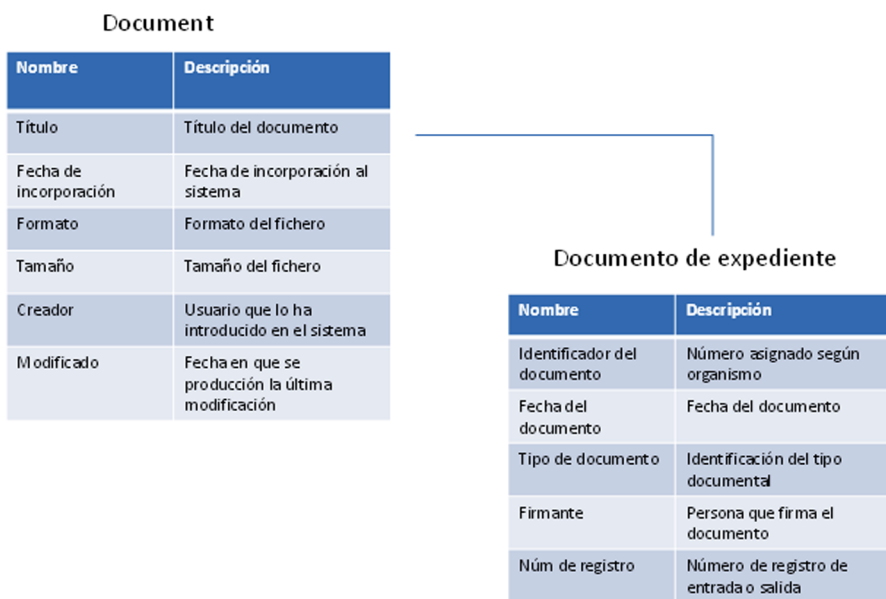
Por último, tendremos que buscar la forma de implementar los elementos de metadatos que no estén ya definidos en el modelo y los tipos de objeto del EDRMS.

2.5.3. Implementar metadatos no definidos en el modelo

Para esta operación los EDRMS proponen dos fórmulas diferentes según las herramientas:

a) La primera es la que proponen herramientas como Filenet, Documentum o Sharepoint entre otras. Los metadatos personalizados se implementan en tipos de objetos que se derivan del tipo de objeto principal. En la siguiente figura vemos cómo el objeto documento de expediente añade una serie de elementos de metadatos, a los que ya tiene por defecto el objeto documento.

Figura 8. Definición figurada de tipos de objeto heredados



Fuente: Elaboración propia

En esta modalidad los distintos atributos o columnas, como se llaman en Sharepoint, pueden ser reutilizados para asignarlos a distintos tipos de objetos.

b) En la segunda, como OpenText o Alfresco, lo que se propone es la creación de un conjunto de elementos de metadatos, que en OpenText se llaman categorías y en Alfresco aspectos, que pueden ser asignadas de distintas formas a

los objetos que vienen por defecto, que tienen sus propios metadatos definidos. En la próxima tabla veremos la definición de una categoría en OpenText, que será añadida a todos los objetos *folder* creados.

Categoría: Titular del expediente

Aplicable a todos los expedientes (múltiple)

Tabla 1. Definición de categoría simulada para OpenText

Name	Type	Description	Value required	Valores
Client name	<i>String</i>	Nombre del titular o cliente de la operación a la que se refiere el expediente	Yes	
ID number	<i>String</i>	CIF o cualquier otro número de identificación asociado con la entidad		
ID client	<i>String</i>	Número identificación interno para cada cliente		En aplicaciones de gestión

Elaboración propia

Los atributos o elementos se definen en cualquiera de las dos modalidades, asignándoles una serie de características, que van desde el tipo de campo, las etiquetas por las cuales se van a reconocer hasta el comportamiento de los valores esperados. Por ejemplo, si son campos de valor requerido, si aceptan valores repetidos, si los valores se escogen de tablas predefinidas o si existe un valor por defecto.

2.5.4. Definición de las formas de cumplimentación

Una de las partes más importantes es determinar cómo se van a cumplimentar estos campos o cómo se alimentan de otras aplicaciones. La cumplimentación manual de los metadatos, que supone que un usuario teclee los valores, debe siempre limitarse a los casos en los que sea imposible hacerlo de otra forma. Por ejemplo, se puede determinar que un atributo de fecha se cumplimenta con los datos de la fecha del sistema, o un atributo de nombre de persona con los datos del usuario que captura el documento en el sistema. Cuando la información se encuentra en otras aplicaciones de gestión, las integraciones pueden resultar más complejas y requerir de programación o desarrollos específicos, pero siempre valen la pena si conseguimos la automatización en la cumplimentación de los metadatos.

Nota

En muchas implementaciones de las herramientas que proponen la creación de distintos tipos de objetos a partir de un objeto padre, por ejemplo en Documentum denominados *doc-types*, se ha entendido que era la forma de implementar los “tipos documentales” tales como se entienden en archivística o en gestión documental. Por ejemplo, si las NTI

proponen una lista de 20 tipos documentales, como resolución, contrato, comunicación, denuncia, etc., estos deben convertirse obligatoriamente en distintos tipos de objetos en el EDRMS. Esta forma de entenderlo se agudiza porque prácticamente todas las herramientas ponen ejemplos de creación de tipos de objetos específicos, conceptos que se asemejan mucho a los tipos documentales de la gestión documental.

Esta aproximación puede utilizarse y tiene sentido cuando realmente nuestro esquema de metadatos o perfil de aplicación quiera establecer elementos de metadatos y comportamientos diferentes para cada uno de estos tipos. Cuando no es así, no es necesaria la proliferación de tipos de objetos que en realidad tienen los mismos metadatos y se comportan de la misma forma. Existen implementaciones que manejan más de 200 tipos de objetos personalizados de este tipo, lo que puede dificultar el control de los mismos.

No existe una decisión única válida para todas las implementaciones a este respecto, pero la que sea debe partir del entendimiento de que estos dos conceptos no son iguales incluso aunque se denominen de la misma manera en determinados entornos.

2.5.5. Documentar el perfil de aplicación

Además de la implementación en la propia herramienta, conviene documentar el perfil de aplicación de forma independiente, convirtiéndose en uno de los elementos de control del *governance framework*. Este perfil de aplicación define las reglas de la organización y las directrices para el uso de los elementos en la aplicación donde se vayan a implementar, proporcionando comentarios y ejemplos para ayudar a la comprensión de cómo hay que hacerlo.

3. La implementación del modelo de acceso

En este apartado estudiamos el modelo de acceso a la plataforma tecnológica, a partir de reflexionar sobre la importancia y los mecanismos para definir la autenticación de usuarios, y el acceso y los permisos sobre los objetos.

3.1. Consideraciones generales

Las posibilidades que la tecnología nos ofrece también determinan cómo implementar el modelo de acceso. En los EDRMS, el acceso y la seguridad se basan en la combinación de usuarios o grupos (sujetos de acceso), con los elementos a los que se quiere dar acceso (objetos de acceso) y los permisos que se asignan.

La identificación de los **objetos de acceso** está estrechamente ligada al modelo de información que se haya definido para el repositorio y que se ha explicado en el apartado anterior.

Los **sujetos de acceso** se deben definir identificando los tipos de usuarios que tendrán los mismos permisos o roles.

Los **permisos** que es posible asignar a cada grupo sobre determinados objetos vendrán determinados por lo que nos ofrece cada una de las tecnologías.

Cada EDMRS ofrecerá sus propias herramientas para la implementación del modelo de acceso. En este apartado trataremos de entender los puntos en común, pero la primera tarea necesaria antes de implementar la tecnología en un EDRMS concreto es estudiar a fondo las posibilidades que nos ofrece.

La implementación del modelo de acceso tiene al menos dos facetas distintas a tener en cuenta:

- La autenticación de usuarios, que verifica la identidad de los usuarios que acceden al EDMRS.
- Los permisos sobre los distintos tipos de objeto en el repositorio, una vez que el usuario está autenticado.

También podemos tener necesidades de incluir en los modelos aspectos más específicos de privacidad sobre un determinado grupo de documentos o incluso sobre determinadas partes de los documentos.

Ved también

Los fundamentos de este proceso se estudian a fondo en la asignatura:

Procesos y herramientas de gestión documental (C. Bustelo Rueda; M. Juncà Campdepadrós, 2013).

3.2. Autenticación y gestión de usuarios

Para la autenticación de la identidad de los usuarios, lo habitual es que los EDRMS se apoyen en un servicio corporativo responsable de toda la autenticación de las identidades. Puede existir una sola fuente de autenticación, como puede ser el *LDAP (lightweight directory access protocol)*, que da servicio a todas las aplicaciones y sistemas, o tener varias fuentes de información establecidas de forma que son sincronizadas de manera distribuida desde el servicio maestro.

En estos directorios corporativos se mantienen los usuarios y sus datos, y también pueden establecerse los grupos de usuarios que se necesitan en las distintas aplicaciones.

Normalmente, estos directorios están gestionados por los responsables de sistemas TIC y son una pieza fundamental en el mantenimiento de la seguridad informática. Sin embargo, no pocas veces nos encontramos conflictos con los departamentos funcionales o responsables de otras aplicaciones sobre las responsabilidades del mantenimiento al día de estos directorios. La implementación de un EDRMS es normalmente una de las ocasiones en la que estos conflictos salen a la luz y afectan a la forma en que se establece el modelo de acceso a los mismos.

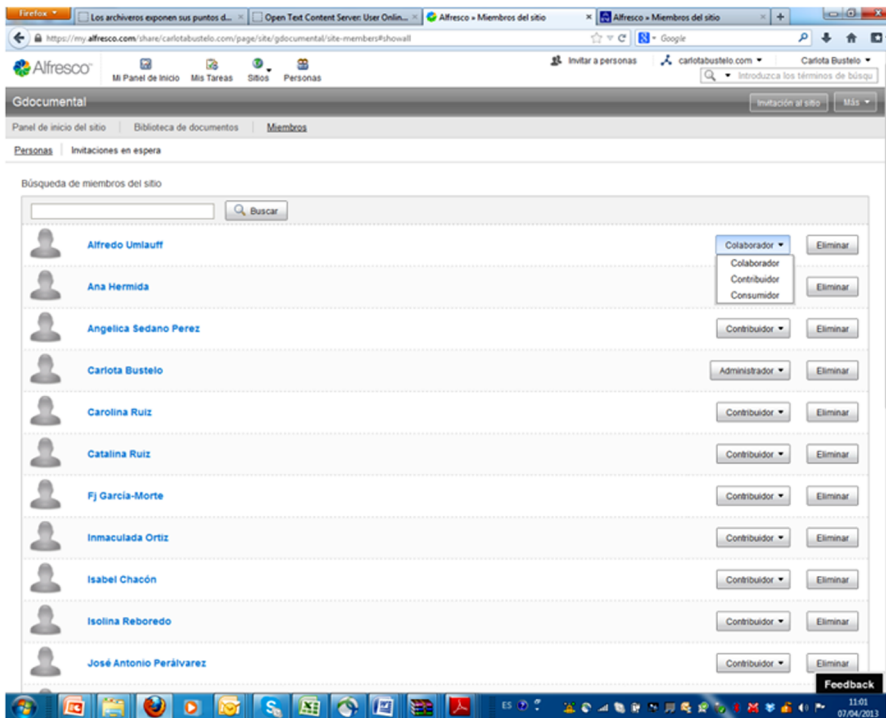
Los escenarios más comunes son los siguientes:

1) El modelo de seguridad que se quiere implementar para el EDRMS se basa en los usuarios nominales que ya están definidos en el directorio corporativo. A partir de estos, necesitamos crear los grupos que se utilizarán en la aplicación del modelo de acceso. Normalmente, la implementación de cada modelo de acceso requiere de la creación de muchos grupos diferentes de usuarios que combinarían los roles que se ejercen con respecto a un conjunto de documentos. Una vez analizadas y entendidas las necesidades de creación de grupos de usuarios por parte de los que tienen que decidirlo, pueden tomarse dos opciones diferentes:

- Los responsables del directorio asumen la tarea de creación y mantenimiento de grupos y se establece el protocolo que permite realizar las altas, bajas y cambios de pertenencia a los grupos de una manera asumible y eficaz.
- El directorio tiene un montón de grupos creados según las necesidades de cada aplicación, pero ninguno de ellos responde a la lógica requerida para el EDRMS. Los responsables del directorio dicen ser incapaces de la gestión y mantenimiento de los grupos en el directorio y, por lo tanto, hay que buscar otra solución. Una de las posibles soluciones es la creación de un solo grupo de autenticación para el EDRMS en el LDAP, mientras que la asignación de usuarios a grupos o roles concretos se realiza en el propio

EDRMS. En la figura 9 vemos un pantallazo de la interfaz que propone Alfresco Share para que el propio usuario desde una interfaz muy sencilla vaya creando los grupos de usuarios asignados a un determinado perfil para un sitio concreto.

Figura 9. Captura de pantalla de aplicación Alfresco en la nube



2) Cuando en la arquitectura de información definida el EDRMS es el repositorio de documentos al que se ha accedido desde distintas aplicaciones de gestión, puede plantearse que la lógica de la seguridad reside no en los propios EDRMS, sino en las aplicaciones que los alimentan y los consultan. De esta forma el usuario que consulta el repositorio siempre es el mismo, lo que se denomina usuario aplicación o usuario genérico. Este usuario aplicación o genérico es el que se encontraría en el LDAP y es el que produciría la autenticación de acceso al LDAP. En este escenario también se pueden producir dos opciones:

- Los usuarios nominales y los grupos se usarían para la autenticación de acceso a las aplicaciones de gestión. De esta forma se facilita la implementación desde el punto de vista tecnológico, pues el EDRMS tendría un modelo de acceso muy sencillo, pero se anularían gran parte de las funcionalidades de los EDRMS, como la posibilidad de acceso directo al repositorio y la creación de las pistas de auditoría.
- La organización desarrolla una capa intermedia de gestión de usuarios y grupos entre los LDAP y las aplicaciones, que permite aislar las tareas de mantenimiento de grupos de las tareas de autenticación. Este tipo de soluciones son válidas para todas las aplicaciones de forma centralizada.

Con respecto a la autenticación, merece una atención especial la posibilidad de dar acceso a los EDRMS a usuarios externos a la organización. En estos casos los EDRMS no son diferentes que el resto de las aplicaciones y, por lo tanto, se deberán seguir las directrices de la organización en este sentido.

Habitualmente representa un problema, pues es difícil mantener usuarios externos en el LDAP, que es la manera común de autenticación. Para permitir este acceso además es necesario que la aplicación EDRMS sea accesible desde el exterior. Una vez que esto es posible, se debe establecer cómo será el modo de autenticación.

En muchos casos se solventa con una aplicación intermedia de autenticación, que resuelve la petición externa (mediante usuario/contraseña, certificado digital o cualquier otro método) y la convierte en un usuario genérico externo del LDAP.

Muchos EDRMS están potenciando sus soluciones en la nube o mixtas para resolver el problema de permitir el acceso a usuarios externos a determinados objetos del repositorio.

3.3. El acceso y los permisos sobre los objetos

En este apartado estudiamos cómo implementar el modelo de acceso. Concretamente, el control mediante listas de acceso, los niveles a los que se implementan y los roles que se pueden determinar.

3.3.1. El concepto de ACL (*access control list*)

En los sistemas EDRMS, el modelo de acceso se implementa sobre las *ACL (access control list)*. Una *ACL* es una lista que comprende las identidades y las autorizaciones específicas de acceso y control para cada identidad sobre el objeto al que están asignadas. Las identidades específicas están determinadas por la forma de autenticación de usuarios. Por lo tanto, la *ACL* está asignada a un objeto específico y está compuesta por distintas entradas que se denominan *ACE (access control entry)*.

Para que un usuario que ya ha sido autenticado en el LDAP acceda a un objeto, es necesario que esté autorizado para efectuar las acciones mediante los permisos establecidos en las *ACL*.

Muchas veces las *ACL* son a la vez otro tipo de objetos, con sus propios comportamientos y permisos. Cada una de las herramientas nos proporcionará las funcionalidades más o menos sofisticadas para la construcción de estas *ACL*.

Una de las partes comunes a todas las herramientas es que las ACL siempre se interpretan de la forma más restrictiva. Por ejemplo, si un usuario individual tiene permiso en la ACL para acceder a un documento, pero a su vez pertenece a un grupo que no lo tiene, el usuario no podrá acceder.

Normalmente todas las herramientas nos ofrecerán la opción de creación de ACL por defecto, para facilitar la implementación de la seguridad.

Por otro lado, las ACL pueden ir cambiando en función de los distintos estados de los documentos. Estos cambios se producen de forma automatizada y pueden estar implementados por defecto o personalizarse. El ACL más habitual de los que vienen por defecto, cuando se implementan los módulos de *records management*, es el cambio en el acceso que se produce en la “declaración de record” y que supone la eliminación de la ACL de todos los permisos de edición para todos los tipos de usuarios.

3.3.2. El nivel donde situar las ACL

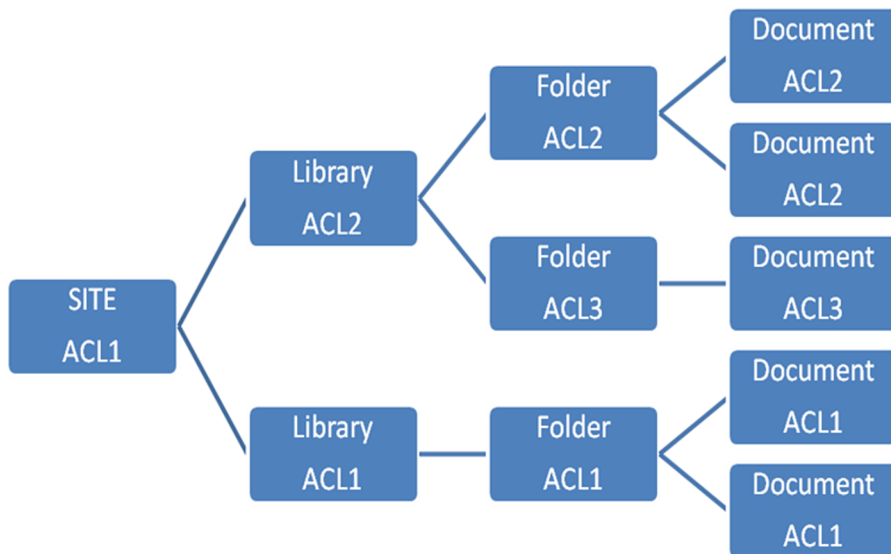
Como hemos dicho, las ACL están asignadas a los objetos, así que la manera más básica de crear una ACL es que el usuario que incorpore el documento en el sistema defina cual es su ACL.

Siendo esta una posibilidad tecnológica de todas las herramientas, no es una práctica aconsejable para implementar un modelo de acceso. Primero, porque es tremendamente tediosa para el usuario y después porque el modelo de acceso se convierte en algo difícil de gobernar y que puede llegar a afectar al rendimiento de la aplicación cuando hablamos de un gran número de documentos.

Por eso todas las herramientas nos proveen de otras herramientas que nos permiten asignar las ACL a otros niveles, de manera que sean heredadas por los objetos dependientes o contenidos.

El caso más claro es la asignación de las ACL a nivel de objetos contenedores, como son los de tipo *folder*, para que sean heredadas a todos los objetos “contenidos” bien sean otros objetos *folder* u objetos tipo *document*. En la figura 10 vemos los niveles donde se puede implementar las ACL en una jerarquía de Sharepoint. La ACL asignada a un *site* (ACL 1) es heredada mientras no se defina otra a un nivel más bajo. Este nivel puede ser el nivel de *library* o biblioteca de documentos (ACL2) o a nivel de un *folder* dentro de la biblioteca (ACL3).

Figura 10. Herencia de ACL en una jerarquía Sharepoint



Fuente: Elaboración propia

Este es un tipo de modelo que encaja muy bien con el modelo de acceso extraído de las buenas prácticas de gestión documental. De hecho es el que ofrece habitualmente cuando se implementan los módulos RM (de *records management*). Una vez que el documento está clasificado en el *file plan*, la ACL que hereda es la de su clase o categoría.

Otra forma posible de implementar la herencia de ACL es a través de la jerarquía de tipos de objetos. Por ejemplo, se asigna una determinada ACL a un tipo de objeto definido que es heredado por todos los objetos que se han definido a partir de esta. Esto supone, por ejemplo, entender que la ACL puede aplicarse a un tipo de objeto específico llamado contrato, de manera que todas las instancias de ese objeto tendrán las mismas ACL. Es decir, los permisos de acceso serán los mismos para todos los contratos. La posibilidad de utilizar esta opción depende totalmente de las decisiones que hayamos tomado con respecto al modelo de información del apartado anterior.

Aunque se utilice la asignación de permisos por herencia, siempre será posible modificar (restringir) los permisos concretamente a un determinado documento o instancia del objeto documento o *folder*. Esto será útil para la aplicación de excepciones, por ejemplo, de confidencialidad, o para limitar los permisos a un solo individuo al que se le ha asignado una tarea, como veremos en el apartado siguiente.

Para decidir cómo los vamos a implementar, deberemos entender cómo funciona la herramienta concreta con respecto a la creación de la ACL. Al menos podemos distinguir:

- Los permisos que se crean por defecto para cualquier objeto. Estos vienen normalmente determinados por el tipo de objeto. En la siguiente figura, vemos los permisos por defecto en OpenText.

Figura 11. Permisos de Open Text Content Server por defecto a cualquier objeto

- **Owner** 👤, the user that has full permissions on the item. By default, the Owner is the user who created the item. However, ownership of an item can be transferred. Depending on your organization's requirements, this user does not necessarily have full permissions on the item.
- **Owner Group** 👤, a group to which you want to assign extra permissions on the item. By default, an item's Owner Group is inherited from its container.
- **Public Access** 🌐, a designation that makes the item available to all users who have the Public Access privilege. Designating an item as Public Access lets you specify a general permissions set that is appropriate for general use. Because Public Access is a privilege that is in effect systemwide, you cannot modify the users included in this group.

Fuente: Open Text Content Server Online Help

- Los permisos heredados que se basan en los permisos del objeto superior que los contiene.
- Los permisos individualizados que se asignan directamente sobre el objeto.
- Los permisos que pueden ser asignados por otros objetos personalizados que algunas herramientas dejan crear y que son especialmente utilizados cuando se quiere controlar el cambio de ACL a través, por ejemplo, de un flujo de trabajo o sobre las distintas versiones existentes.

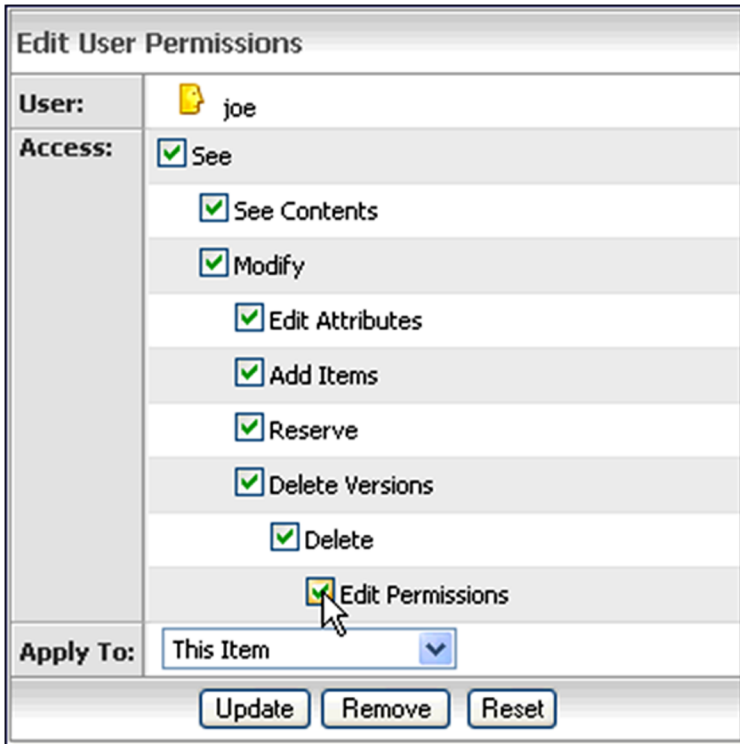
En algunas herramientas se ofrece además una funcionalidad más sofisticada que permite el establecimiento de permisos mediante un valor específico de un metadato, que actúa en conjunto con la ACL. Por ejemplo, si el objeto tiene un metadato con valor “confidencial”, solo se dejará el acceso al grupo específico de individuos que son los que tienen acceso a los documentos así marcados.

3.3.3. Los roles preestablecidos o la personalización de permisos

Los **permisos** suponen las acciones que los usuarios identificados pueden realizar sobre los objetos. Cada herramienta tendrá unos permisos específicos que pueden asignarse sobre distintos tipos de objetos. La granularidad de los permisos dependerá de cada herramienta, siendo habitual que los permisos puedan agruparse.

En la figura 12 vemos cómo en OpenText, determinados por la forma de autenticación de usuarios, los permisos que se aplican sobre un determinado objeto se anidan, pudiéndose especificar o personalizar para cada usuario o grupo. Así, bajo el conjunto de permisos “See” se anidan otros más granulares y específicos. Esta forma de asignar los permisos impide contradicciones, ya que determinados permisos son necesarios como base para poder tener otros. Por ejemplo, no se puede borrar un documento que no se puede ver.

Figura 12. Captura de pantalla de permisos anidado en OpenText



Los permisos pueden asignarse sobre los distintos objetos directamente, como hemos visto en el ejemplo anterior, ligándolos con los usuarios o grupos de usuarios; o asignarlos en base a una serie de roles que se ejercen sobre un entorno determinado.

En algunas herramientas nos vamos a encontrar roles predefinidos que están implementados por defecto. Como ejemplo, vemos en la figura 13 los roles predefinidos en Alfresco.

Figura 13. Roles predefinidos en Alfresco

	Owner	Coordinator	Collaborator	Contributor	Editor	Consumer
Ver espacio de invitado	X	X	X	X	X	X
Ver contenido	X	X	X	X	X	X
Copiar Contenido	X	X	X	X	X	X
Previsualizar contenido en plantilla	X	X	X	X	X	X
Ver propiedades del contenido	X	X	X	X	X	X
Comprobar contenido en el espacio de invitado	X	X	X		X	
Solicitar contenido a un espacio diferente	X	X	X		X	
Update/edit content created by other users	X	X	X		X	
Actualizar / editar el contenido creado por otros usuarios	X	X	X		X	
Editar los debates existentes	X	X	X		X	
Crear y añadir nuevo contenido	X	X	X	X		
Cortar o borrar contenido de otros usuarios	X	X				
Crear espacios de hijos en el espacio de invitados	X	X	X	X		
Ver normas de contenido	X	X	X	X		
Solicitar contenido al mismo espacio	X	X	X			
Contribuir a discusiones existentes	X	X	X	X		
Invitar a otros	X	X				
Iniciar nuevo tema en una discusion	X	X	X	X		
Borrar contenido creado por otros usuarios	X	X				
Mismos derechos que el propietario	X	X				
Tomar posesión de contenidos	X	X				
Crear reglas en espacios	X	X				

Normalmente, en todos los EDRMS existen unos permisos especiales que, aunque se pueden llamar de manera diferente en cada herramienta, mantienen la misma lógica:

- Uno es para la asignación de permisos a “todo el mundo”. En algunas herramientas se llama el grupo de usuarios autenticados, y se suele utilizar para dar permisos de lectura a determinados objetos. Funciona como un grupo lógico al que pertenecen todos los usuarios que puedan acceder a la aplicación. En algunas herramientas, como SharePoint o Alfresco, dar estos permisos sobre un determinado objeto se hace mediante la funcionalidad “Publicar”.
- Otro son los permisos especiales del “Owner” o creador del objeto. Normalmente tiene todos los permisos sobre los objetos que ha creado o *full control*. Este permiso se crea por defecto y puede ser cambiado modificando las ACL correspondientes.

Cada herramienta nos ofrecerá unas interfaces específicas que nos permitirán la creación de las ACL correspondientes a nuestro modelo de seguridad con una mayor o menor facilidad. Cada una de ellas funciona de una manera diferente, aunque todas ellas buscan la mayor facilidad en la implantación.

Antes de proceder a la creación de las ACL, conviene que, utilizando el lenguaje de la herramienta específica, hayamos construido las tablas de doble o triple entrada, que recojan el modelo de acceso acordado. En la tabla 2 encontramos un ejemplo de un modelo construido sobre Documentum.

Tabla 2. Modelo de acceso sobre Documentum

Nivel	Roles	Propietario (OWNER)	Responsable de la serie	Responsable del proceso	Responsable del proceso aprobador	Superiores jerárquicos	Resto (WORLD)
SERIE (folder)		DELETE (si está vacía)	WRITE	BROWSE	BROWSE	BROWSE	BROWSE
– Abierta		DELETE (si está vacía)	WRITE Change location Change state	WRITE	WRITE	READ	NONE
– Cerrada		READ	READ	READ	READ	READ	NONE
DOCUMENTOS							
– Borrador		DELETE	NONE	NONE	NONE	NONE	NONE
– Creado		DELETE	READ Change location	WRITE Change state	WRITE	BROWSE	NONE
– Revisado		WRITE	READ Change location	WRITE Change state	WRITE Change state	READ	NONE

Fuente: Elaboración propia

Nivel	Roles	Propietario (OWNER)	Responsable de la serie	Responsable del proceso	Responsable del proceso aprobador	Superiores jerárquicos	Resto (WORLD)
– Definitivo		VERSION Con carpeta cerrada READ	READ Change location	VERSION Con carpeta cerrada READ	VERSION Con carpeta cerrada READ	READ	NONE

Fuente: Elaboración propia

Actividad

Partiendo de este modelo de acceso hagamos una propuesta de cómo implementar los grupos de usuarios y de los niveles a los que se debe implementar las ACL.

3.4. Técnicas especiales de seguridad: la encriptación y la anonimización

En lo que se refiere al modelo de acceso, pueden aparecer otras facetas relacionadas con la privacidad a tener en cuenta en la implementación de un EDRMS.

Esto se produce cuando algún tipo de información recogida en los objetos documentales requiere de altas medidas de seguridad, como la utilización de la técnica de la criptografía más allá de la utilización habitual en las comunicaciones de la información que se traslada de un lugar a otro (*secure sockets layer, SSL*). Cuando la utilización de estas técnicas sea necesaria por la implantación de medidas de seguridad, hay que asegurarse de la adecuada gestión de la claves de desencriptación.

Por otro lado, es frecuente que al implementar el modelo de acceso a grandes repositorios de documentos, se incluyan requisitos de anonimización de documentos. Fundamentalmente consiste en la ocultación de determinados datos del contenido de un documento, con el ánimo de proteger informaciones confidenciales (habitualmente datos de carácter personal).

La implementación de la anonimización de documentos incluidos en un repositorio EDRMS todavía no ha sido resuelta de manera satisfactoria. Algunos EDRMS ofrecen productos de terceros para actuar sobre los documentos que gestionan. En inglés se llaman productos de *redaction*.

Ved también

Normalmente, para esto se utilizan visualizadores que estudiaremos en el apartado de "Búsquedas y visualizadores".

4. *Workflow* y ciclo de vida de los documentos

En este apartado estudiamos la traslación del ciclo de vida de los documentos en la plataforma tecnológica. Concretamente, tenemos en cuenta los flujos de trabajo y su relación con los documentos, el versionado de estos, y los eventos o reglas que se deben contemplar en la aplicación, para la correcta gestión de los documentos.

4.1. Consideraciones generales

En la implementación de un EDRMS también tendremos que decidir cómo utilizar las funcionalidades que estos incluyen para gestionar los distintos estados o etapas de los documentos, así como las posibilidades de la automatización de flujos de trabajo o eventos relacionados con los mismos.

La utilización o no de estas funcionalidades depende en gran parte del alcance y los objetivos de la implementación. Fundamentalmente, los tendremos que utilizar cuando el alcance de nuestro proyecto de gestión documental incluye la etapa de preparación de los documentos y especialmente, cuando en la creación de los documentos interviene más de una persona. Antes de comenzar debemos tener en cuenta que la concepción de los estados y el ciclo de vida se prolongan en aplicación de las políticas de disposición o retención propias de los módulos de *records management* y que veremos en otro apartado específico.

Es quizás en estas funcionalidades donde las herramientas EDRMS se diferencian más unas de las otras. Incluso podemos encontrar un cierto solapamiento con otras herramientas denominadas *BPM* (*business process management*), que a veces también constituyen módulos específicos de las plataformas ECM.

En este apartado estudiaremos las claves para implementar estas funcionalidades suponiendo que se ha incluido en el alcance del proyecto de implementación dentro del propio EDRMS y no mediante la utilización de otras herramientas externas. En este último caso, muchas consideraciones también serían válidas, pero teniendo en cuenta las necesidades de integración y las funcionalidades propias de las herramientas a integrar.

4.2. Los ciclos de vida de los documentos

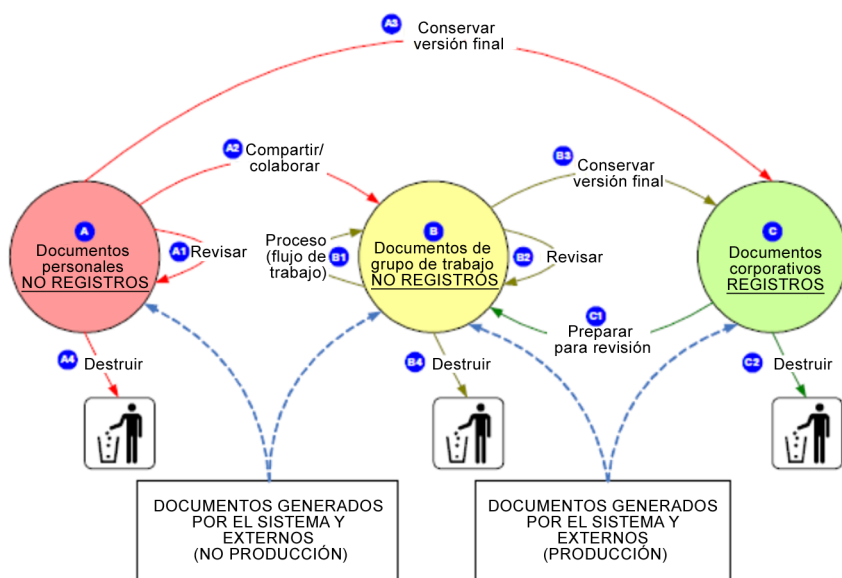
Algunas herramientas EDRMS nos van a permitir definir un conjunto de reglas de negocio para los documentos, que indican los distintos estados por los que pasa un documento desde su captura en el sistema.

Todos los EDRMS que incluyen funcionalidades de *records management* tienen al menos el estado de *record* implementado por defecto; pero todos los demás estados hay que definirlos según las necesidades de la organización.

No existe ninguna regla para la definición de estos estados, que será posible y más sofisticada dependiendo del entorno tecnológico. Lo habitual será que estén en relación con las aplicaciones de gestión que crean los documentos con los distintos procesos de revisión y aprobación de documentos.

Un ejemplo de definición de diferentes estados puede ser: Borrador, Aprobado Pendiente de firma, Definitivo.

Figura 14. Representación de un posible ciclo de vida de los documentos



Fuente: IBM FileNet Content Manager Implementation Best Practices and Recommendations

En la figura 14 se presenta un diagrama de tres estados. En cada uno de los estados solo están permitidas unas determinadas acciones. En los dos primeros (documentos personales y documentos de grupo) un documento puede ser revisado y ser promovido al siguiente estado o finalizar su ciclo de vida. En el último (documento corporativo) un documento puede ser devuelto al estado anterior.

El ciclo de vida o reglas, como algunos EDRMS lo llaman, puede estar implementado como un objeto asociado siempre a un tipo de objeto determinado, normalmente derivado del tipo de objeto *document*.

El ciclo de vida de un documento es la progresión de estados, pero no la definición de cómo y cuándo se pasa de uno a otro. Esto último se suele hacer en los entornos más sofisticados, con los flujos de trabajo o de revisión de documentos que veremos en el siguiente apartado.

Los distintos estados de los documentos están normalmente asociados a distintas condiciones de acceso. Esto se produce normalmente porque un paso de un estado a otro ejecuta una serie de acciones, una de las cuales es el cambio de la ACL, estudiado en el apartado anterior.

En las implementaciones de un EDRMS en las que se estén usando mecanismos de firma electrónica se deberían usar los estados en relación con el momento en el que se procede a la firma. Esto nos permitiría incluir el mecanismo de firma de documentos en un estado específico y controlar de esta forma el uso del mismo.

El concepto de ciclo de vida no se aplica más que a los objetos tipo *document*. En las herramientas en que *récord* es un tipo de objeto diferente, el ciclo de vida no es aplicable. Por naturaleza un *récord* es inmutable y por lo tanto, genera las versiones o estados que componen un ciclo de vida. En algunas implementaciones, especialmente en la Administración pública española, se implementa el concepto de expediente electrónico simulando una especie de ciclo de vida de las carpetas, que al menos tiene dos estados, abierto y cerrado. El estado de expediente cerrado implica la imposibilidad de incluir más documentos en el mismo.

Nota

Cuando se establecen piezas que facilitan la firma de documentos, como son los portafirmas electrónicos, debemos asegurarnos de que los documentos firmados acaban en el repositorio en su lugar correspondiente y no en el propio portafirmas.

4.3. Los flujos de trabajo y su relación con los documentos

En muchos EDRMS se incluyen flujos de trabajo básicos. En principio se distinguiría entre flujos de trabajo básicos asociados a los documentos y otros flujos de trabajo más sofisticados, que no solo se asocian a documentos sino que podrían automatizar procesos más complejos, que encadenan unas decisiones con otras.

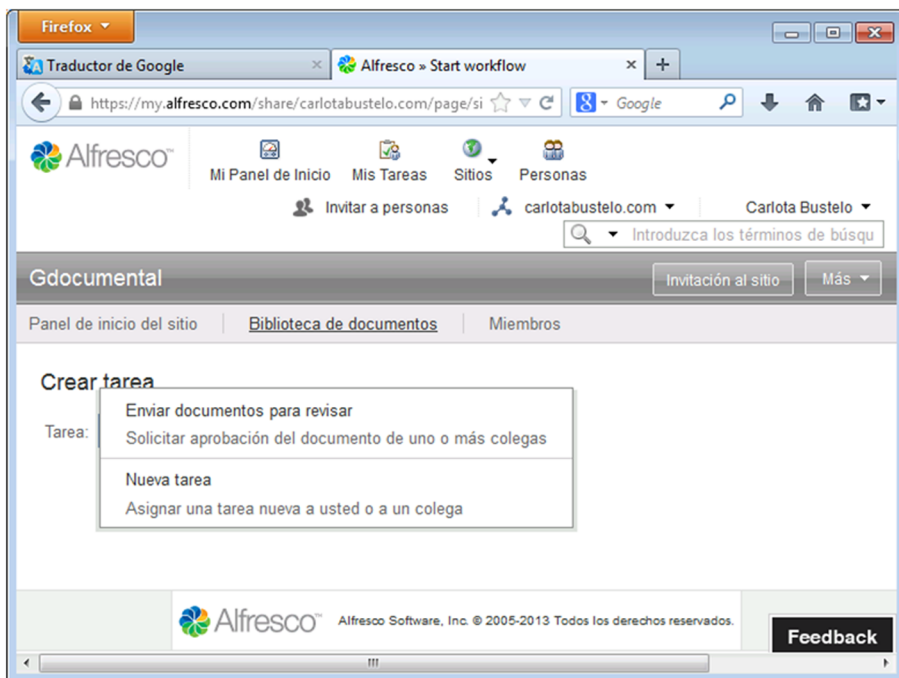
Los flujos básicos que suelen venir implementados en un EDRMS son, al menos:

- Revisión de un documento por una persona.
- Revisión de un documento por varias personas.
- Aprobación de un documento.

En la implementación de un EDRMS, lo primero que tendremos que estudiar es el funcionamiento real de los flujos que vienen implementados por defecto, para determinar si es necesaria una personalización o incluso si es preferible la utilización integrada de otra herramienta BPM utilizada en la organización.

En este apartado estudiaremos las formas más básicas de los flujos en los EDRMS y apuntaremos cuándo pueden existir mayores funcionalidades que solo tienen determinadas herramientas. En la figura 15 se muestran los flujos básicos ofrecidos por Alfresco en la nube, y que solo consisten en mandar un documento a revisar o enviar una tarea.

Figura 15. Captura de pantalla que muestra los dos tipos de flujos-tareas ofrecidos por la versión gratuita de Alfresco en la nube



La forma más básica de un flujo de trabajo en un EDRMS se suele llamar tarea. La idea general es que se puede asignar una tarea a un usuario o a varios. Las tareas son normalmente otro tipo de objeto, con unos datos y comportamientos predefinidos.

Las tareas pueden asociarse a otros tipos de objetos. Cuando se aplican para aprobación o revisión de documentos, será sobre los objetos de tipo *document*. Para los récords los flujos y tareas que suelen aplicarse son las relacionadas con la disposición (aprobación de la eliminación o de la transferencia a otros repositorios).

Las tareas se asignan a un usuario y grupo de usuarios y mediante la cumplimiento de una serie de atributos de la tarea se establecen los parámetros de la misma. Uno de los atributos que son siempre imprescindibles es la fecha en que debe estar realizada la tarea.

En la tabla 3 se recogen los atributos por defecto de las tareas en Open Text.

Tabla 3

Attribute	Description
Assigned To	The user or group to whom the Task is assigned. You do not have to assign a Task, but if you do, that user or group must have at least the Write permission.
Start Date	The date when the Task was created or the date work should begin.
Due Date	The date when the Task should complete. You can create a Task without specifying a due date. If a Task's due date has passed, the Task's Due Date attribute appears in red on the Task, Task List, and Assignment pages.
Priority	You can specify a Task as high, medium, or low priority.

Attribute	Description
Status	You can specify a Task as pending, in process, issue, on hold, completed, or cancelled. If the Task is completed, it still displays on the Task List.
Milestone	The Milestone that is associated with the Task.
Instructions	An optional attribute that allows you to specify instructions for completing the Task.
Comments	An optional attribute that allows assignees to add comments about the Task.
Attachments	This attribute appears in a separate section and is available only when a Task has documents or other items attached to it, such as Polls, Shortcuts, and URLs.

Cuando se quiere aplicar una tarea a un documento, este aparecerá normalmente como un adjunto a la misma y lo normal es que el documento quede marcado como que está siendo revisado y que posteriormente se puedan consultar las tareas asociadas al mismo.

Las tareas llegan habitualmente a un buzón de tareas, que puede estar sincronizado o no con el correo electrónico. Para finalizar la tarea se nos pueden ofrecer varias opciones, que en los casos elementales de los que estamos tratando pueden ser dos:

- Aprobar
- Rechazar

En ambos casos, el que ha asignado la tarea recibe comunicación del resultado de la misma en su propio buzón.

Para plantear la implementación o personalización de las tareas debemos tener en cuenta los siguientes aspectos:

1) La tarea y el documento adjunto suelen ser dos instancias separadas, así que pulsar el botón *Aprobar* en una tarea de revisión de documento no implica obligatoriamente que se hayan hecho modificaciones en el mismo. El control de las relaciones entre el objeto *Documento* y el objeto *Tarea* se hacen con herramientas más sofisticadas de flujos de trabajo.

2) Cuando se ofrecen revisiones o tareas que se asignan a más de una persona hay que entender cuál es el funcionamiento ofrecido por la herramienta:

- Pueden existir revisiones **en serie**, en las que se escoge el orden de los revisores de forma predeterminada y las tareas se encadenan unas con otras. Esta funcionalidad no es común en las herramientas básicas, que lo que proponen es una asignación de una tarea a una persona y cuando acabe, a otra y así sucesivamente, pero siempre pasando por la persona que las ha asignado.

- Pueden existir revisiones **en paralelo**, que se asignan a varias personas. En este caso lo normal es que la tarea “se la quede” el que primero la ejecuta, desapareciendo como tareas pendientes de los demás. Aunque también se puede especificar el porcentaje de personas que deben dar su aprobación, por ejemplo. Si se lo mandamos a cuatro y decimos que necesitamos un 50%, cuando dos hayan completado la tarea es cuando se dará por finalizada.

En las herramientas más sofisticadas se pueden desencadenar acciones cuando se completa una determinada tarea (por ejemplo, convertir el documento en PDF) o controlar el acceso (cambiar la ACL) a un documento que está asignado a un flujo mientras este se completa.

Este tipo de implementaciones de flujos personalizados requiere de un estudio de los requisitos en profundidad, que siempre acaban resultando mucho más complejos de lo que parece a primera vista. En estos casos además se produce siempre una dificultad en poder distinguir entre los que sería flujos de trabajo documentales y lo que sería la automatización de un flujo de trabajo en el que se utilizan documentos.

En la implementación de funcionalidades de flujo de trabajo habrá que tener un esquema claro de cómo se van a ejecutar dichos flujos y quién tendrá los permisos necesarios para iniciarlos.

4.4. El versionado

En la implementación de un EDRMS también se deben tomar decisiones con respecto a cómo se van a tratar las versiones de los documentos. Las funcionalidades de versionado son básicas en los EDRMS, y están muy relacionadas tanto con los estados como con los flujos de trabajo. En muchas herramientas se nos permitirá habilitar o no para una determinada parte del repositorio las funciones de versionado, teniendo en cuenta que en cuanto los documentos se hayan declarado como *records* se perderán. Mientras los documentos no sean definitivos, cada vez que se modifique un documento se puede optar por:

- que la nueva versión sustituya a la anterior;
- guardar un histórico de versiones.

Esta segunda opción solo tiene sentido cuando existe una razón importante para mantener las versiones o cuando estamos implementando espacios de colaboración entre personas.

Las versiones de un documento suelen ser en los repositorios un tipo de objeto especial que solo puede existir asociado a un tipo de objeto *document*.

La implementación del versionado suele ofrecer la posibilidad de distinguir entre versiones menores y mayores. Normalmente esta distinción no tiene más importancia que el número asignado automáticamente a las nuevas versiones.

Dependiendo de la herramienta podrán o no crearse como nuevas versiones documentos con formatos diferentes al original o puede existir un control del tipo de fichero que puede subirse como una nueva versión.

Es posible que en determinados entornos se hable de distintas versiones definitivas de un documento. Puede ser el caso de los procesos normativos, en los que se deben guardar las distintas versiones del texto legislativo que corresponden a las distintas fases de su tramitación o las distintas versiones publicadas de un determinado procedimiento. Aunque utilicemos la misma palabra, *versión*, debemos entender que su naturaleza no es la misma que las posibles modificaciones o mejoras, por ejemplo, de un informe antes de elevarse como definitivo. Estas serían las que llamamos versiones definitivas, que en algunas herramientas sofisticadas existen objetos específicos para este otro tipo de versiones llamados ediciones, generaciones u otras posibilidades.

También en algunas herramientas existe otro tipo de objeto similar a una versión que se suele llamar *rendition* y que se asocia a versiones del documento en un formato que no es el original.

Como se ha dicho, todos estos objetos son dependientes de un objeto *document* y no existen por sí solos. La posibilidad de crear este tipo de objetos a partir de uno ya existente va normalmente asociada a los permisos asignados a cada usuario. La tarea de implementación consiste fundamentalmente en construir los casos de uso en los que estos objetos van a ser utilizados o no y asignar los permisos correspondientes.

4.5. Los eventos o reglas

Determinadas acciones sobre los documentos pueden ser automatizadas sin necesidad de intervención explícita de un usuario del sistema. Este tipo de acciones se pueden identificar como eventos, acciones o reglas.

Las reglas o eventos se componen normalmente de:

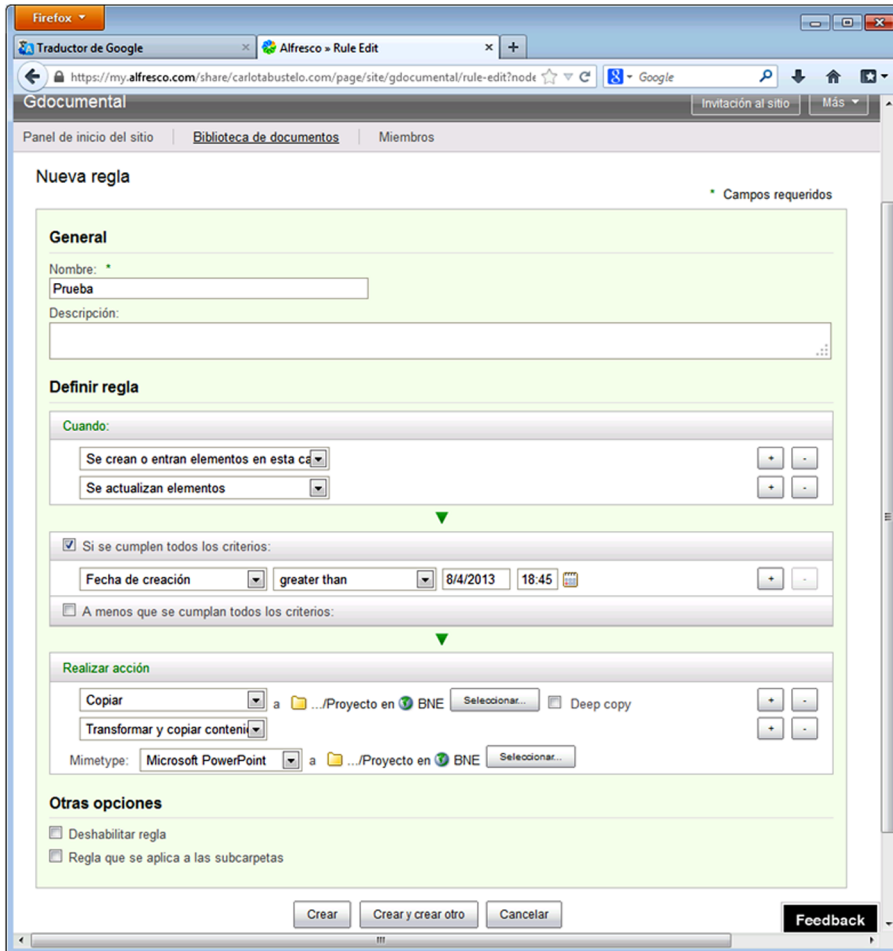
- Un *trigger* o detonante, que es algo que sucede en el repositorio que da lugar a que se ejecute la acción.
- Unas condiciones para que se produzca, que suelen estar basadas en los valores de los distintos atributos de un objeto.
- La acción en sí misma, que se aplica sobre un objeto concreto.

Las acciones siempre pueden realizarse de forma “manual” en el repositorio, pero su inclusión dentro de una regla o evento permite que se ejecuten de una forma automática.

En la figura 16 vemos una captura de pantalla de la interfaz de la versión gratuita de Alfresco en la nube para crear reglas. En este caso las reglas que se pueden crear están muy limitadas pero nos pueden dar una idea de cómo se crearían y aplicarían. En este caso concreto, la regla solo se aplica sobre una carpeta concreta:

- Como detonador se puede elegir entre cuándo se crean documentos en la carpeta, cuándo se actualizan o cuándo se eliminan.
- Las condiciones se escogen de entre los distintos atributos de los documentos, pudiendo poner valor “igual a”, “mayor que” o “menor que” en los casos de atributos numéricos (por ejemplo, fecha del documento); “comienza con”, “contiene”, “termina con”, o “igual a” en los casos de atributos de texto (por ejemplo, título del documento); y la elección de un valor en los casos de listas de valores prefijadas (por ejemplo, formato del documento).
- Las acciones permitidas son en este caso Copiar, Mover, Transformar y copiar contenido, Transformar y copiar imagen. Según el que se escoja hay que indicar qué formato se transforma y a dónde se mueven o copian los documentos.

Figura 16. Pantallazo de cómo se define una regla en la versión gratuita de Alfresco en la nube



Los eventos y reglas son la base sobre las que se asientan las funcionalidades de ejecución de políticas de retención o disposición que veremos en un siguiente apartado.

La combinación de todas las funcionalidades de estados, flujos de trabajo, versiones y acciones conjuntamente pueden ser capaces de la construcción de escenarios muy automatizados en la gestión de documentos. Lo importante antes de empezar a construirlos es tener clara su necesidad, cómo se van a utilizar y el beneficio de su puesta en marcha en la organización.

5. La gestión de los documentos físicos en los EDRMS

En este apartado describimos los objetos relacionados con los documentos físicos y las funcionalidades relacionadas con los objetos físicos.

5.1. Consideraciones generales

La gestión de los documentos físicos, es decir, de los documentos en papel u otros formatos en los EDRMS, está casi siempre ligada a los módulos de *Record Management*.

En muchos casos, previamente a la implantación de un EDRMS, los documentos físicos han sido gestionados mediante aplicaciones clásicas de bases de datos relacionales, e incluso con herramientas de gestión de archivos altamente especializadas.

El intento de utilizar los EDRMS como sistemas de gestión de documentos físicos viene del reconocimiento de la existencia de entornos híbridos, donde documentos en papel se gestionan al mismo tiempo que los documentos electrónicos, y del requisito de los profesionales de la gestión de documentos de unificar entornos. En esta lógica, por ejemplo, se esgrimen argumentos de que manteniendo los dos entornos separados se debe mantener el cuadro de clasificación o *file-plan* en dos entornos distintos, con la consiguiente pérdida de efectividad y el aumento de la posibilidad de errores.

Sin embargo, la diferencia de enfoque y modelo de información es tan distinta a las herramientas tradicionales de gestión de archivos que muchos profesionales se sienten desilusionados cuando ven lo que un EDRMS puede ofrecer con respecto a la gestión de documentos físicos. Por ejemplo, en las prestaciones para la gestión de los espacios o el préstamo de materiales.

En los EDRMS que tratan documentos físicos se han intentado equiparar los objetos utilizados para la gestión de documentos electrónicos con los de objetos físicos. En el siguiente apartado estudiaremos cuales son estos objetos y también las funcionalidades específicas que suelen acompañarlos.

5.2. Los objetos relacionados con los documentos físicos

En el modelo de información habitual de los EDRMS, los objetos que se utilizan para la gestión de los objetos físicos tienen que ser, a la fuerza, diferentes de los que se utilizan para los objetos de documentos electrónicos.

Empecemos por el equivalente del objeto *document* que hemos estudiado anteriormente. La primera diferencia claramente definitoria es que este nuevo objeto no tiene un fichero adjunto, y además, obligatoriamente debe tener unos atributos que indiquen dónde está la copia física de ese documento.

La mayor parte de los EDRMS ofrecen este tipo de objetos llamándolos *physical objects*, *non-electronic document* o *physical record*. En realidad son objetos que solo contienen los metadatos que apuntan al objeto físico (documento en papel, vídeo, fotografía, cinta, CD, etc.). Para un mejor entendimiento vamos a llamarle *objetos tipo documentos físicos*.

Los objetos de documentos físicos pueden estar contenidos en objetos de tipo contenedor, que serían los equivalentes a los objetos *folder* que ya hemos estudiado. Sin embargo, estos objetos contenedores en general deben representar también entidades del mundo real. Es decir un *physical folder* debe responder a que existe en realidad una carpeta física. Esto hace que su forma de comportamiento no pueda seguir la misma lógica que en el caso de los documentos electrónicos, por ejemplo, un objeto documento físico solo puede incluirse en una sola *physical folder*.

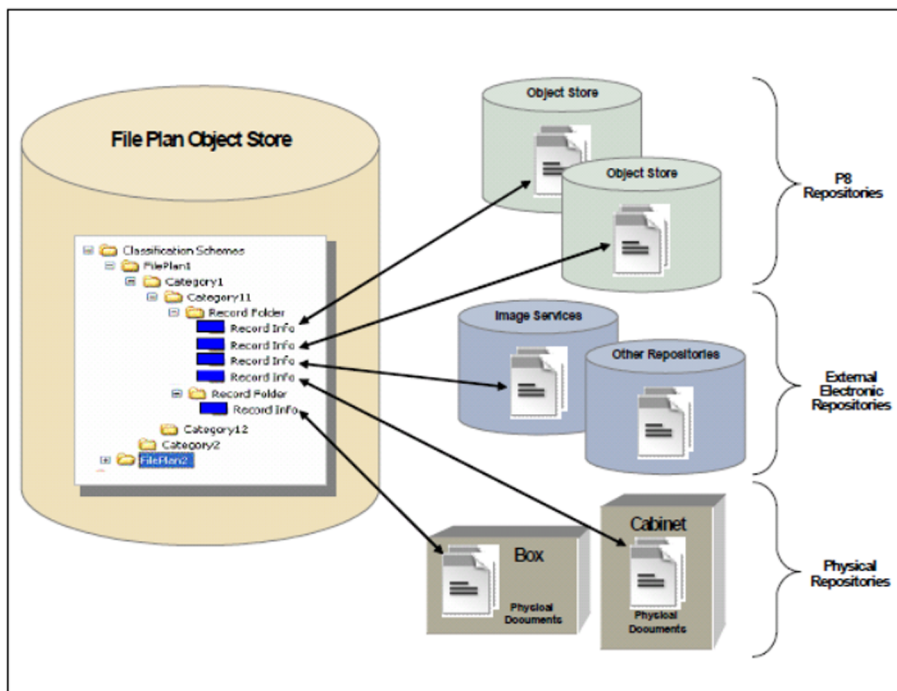
Como el mundo real tiene objetos contenedores de documentos que van más allá de las carpetas, para la gestión de los documentos físicos los EDRMS tienen que proponer nuevos objetos tipo contenedor. El más común es el de caja o *box*, que representa las entidades del mundo real donde habitualmente se almacenan y gestionan los documentos físicos. La mayoría de los EDRMS propone que estos objetos tipo caja no necesariamente tengan que contener objetos tipo documento físicos si la organización no gestiona más que cajas, incluyendo como atributo del objeto caja la descripción del contenido de la misma.

Esta lógica de contenedores puede seguir extendiéndose, y pueden existir objetos que contienen cajas y que en el mundo real corresponderían a depósitos o localizaciones.

Abordar la gestión de documentos físicos en un EDRMS obliga a representar la estructura física en un árbol de objetos jerarquizados, que muchas veces choca con la aproximación utilizada en las bases de datos relacionales, donde la caja, por ejemplo, simplemente se gestionaba con un número secuencial asignado a los documentos o expedientes que contenía.

Sin embargo, estaría justificada por la utilización de unas mismas herramientas de control, como el cuadro de clasificación o *file plan*. En la figura 17 vemos la representación gráfica de la visión sobre este tema de IBM, con su herramienta Filenet P8.

Figura 17. Organización de los repositorios



Fuente: Understanding IBM FileNet Records Manager

Aunque algunas herramientas, como por ejemplo la mencionada FileNet P8, proveen de la posibilidad de contenedores híbridos o mixtos (que admiten objetos físicos y electrónicos), la verdad es que no tienen mucha utilidad puesto que los procesos que se aplican sobre los mismos, especialmente los de disposición que veremos en un apartado más adelante, son obligatoriamente diferentes.

En otras herramientas se separan completamente los contenedores de objetos físicos de los contenedores de objetos electrónicos, incluso gestionándose desde espacios diferentes.

Y en otras, como Sharepoint nativo, no se gestionan objetos físicos, teniendo que recurrir a productos de terceros que sí lo hacen.

5.2.1. Funcionalidades relacionadas con los objetos físicos

Además de la asignación de calendarios de disposición, que veremos en un apartado posterior, y que se aplican tanto a los tipos de objetos de documentos electrónicos y de documentos físicos, existen funcionalidades específicas relacionadas con los objetos físicos que los EDRMS alcanzan con mayor o menor acierto. Se pueden distinguir tres tipos de funcionalidades:

- Las relacionadas con el préstamo y devolución de los documentos físicos.
- Las relacionadas con transferencias de documentos físicos.
- Las relacionadas con la gestión de los espacios físicos y unidades de instalación.

Las funcionalidades básicas de **préstamo** que ofrecen los EDRMS implican como mínimo lo siguiente:

- Poder marcar que el documento está prestado y poder desmarcarlo cuando es devuelto.
- Que un usuario que está consultando un objeto físico pueda pedir el préstamo del mismo.

A partir de estas dos funcionalidades básicas, algunas de las herramientas pueden ofrecer el control de usuarios, de los períodos de préstamo, los préstamos vencidos y el seguimiento de los préstamos de cada objeto. Incluso en algunos casos puede permitir los préstamos múltiples y avisos de plazos excedidos vía correo electrónico. Normalmente, estas funcionalidades son ofrecidas mediante una combinación de atributos asociados a los objetos prestados.

Para implementarlas hay que tener en cuenta cómo es la realidad física de los documentos y las responsabilidades con respecto a la custodia de los mismos. Esto debe traducirse en distintos permisos sobre los objetos físicos, que pueden complicar el modelo de seguridad. Lo más importante cuando estas funcionalidades se quieren implementar es buscar la fórmula que permita que la realidad virtual esté siempre sincronizada con la física. Por ejemplo, que no tenga posibilidades de marcar un documento como devuelto más que la persona responsable de la custodia física de los mismos.

Una propuesta interesante hecha por alguna de las herramientas es aprovechar estas funcionalidades para pedir “copias digitalizadas” de los documentos físicos. La idea central es conseguir un método para ir digitalizando un fondo físico en base a las demandas de los usuarios.

Las funcionalidades relacionadas con las **transferencias** de objetos físicos pueden ser ofrecidas como una función especializada o no según las herramientas. Cuando lo hacen, el mecanismo ofrecido se basa en la posibilidad de ir asignando objetos físicos a otro tipo de objeto “transferencia” o “lista de transferencia”, que permite preparar una transferencia física y confirmarla cuando llega a la siguiente ubicación. Al producirse este último paso, se dará el cambio automático de localización.

Las funcionalidades relacionadas con la **gestión de los espacios físicos** son muy poco comunes en los EDRMS, a no ser que se ofrezcan por productos terceros. Serían las que nos permitirían la gestión de los depósitos, asignación automática de espacios, signaturado automático e impresión de etiquetas.

La implementación de las funcionalidades de gestión de documentos físicos no suele ser el foco principal de la implementación de un EDRMS. Cuando se hace conjuntamente con la implementación de la gestión de los documentos electrónicos, debe tenerse en cuenta que puede constituir un proyecto en sí mismo y no una cuestión subsidiaria con respecto a los documentos electrónicos.

6. Aplicación de las políticas de retención y disposición

En este apartado estudiamos la importancia de incorporar en la plataforma de gestión de documentos los siguientes aspectos: políticas de retención y disposición. Para ello, primero hacemos una breve introducción al tratamiento que las distintas aplicaciones hacen al concepto de retención, después describimos conceptos clave, después estudiamos los aspectos que deben tenerse en cuenta para su implementación, y, por último, temas relacionados con la preservación.

6.1. Consideraciones generales

La posibilidad de automatizar la implementación de las políticas de retención y disposición de documentos en un EDRMS está casi siempre asociada al módulo de *records management*. Sin embargo, es una funcionalidad a la que se ha ido otorgando cada vez más importancia y en algunas herramientas se ha incluido como funcionalidad de base; ejemplos de esto son Sharepoint y Documentum.

Uno de los casos significativos es el de Sharepoint, que las presenta como una de las mejoras de la versión 2013.

También podemos ver que Documentum oferta unos “Retention Policies Services” aparte de la “rigidez” de un proceso formal de *records management*, lo que quiere decir, sin necesidad de comprar el módulo de *records management*.

Independientemente de cómo el software ofrezca la funcionalidad de aplicar las políticas de retención, antes de poder implementar cualquier tipo de políticas de retención o de disposición, es necesario que la organización haya establecido previamente cuáles son estas políticas y cuál es el procedimiento formal para definir las y aprobarlas. Es este enfoque el que debe determinar la implementación de las políticas en el EDRMS, y no al contrario.

Las buenas prácticas en gestión de documentos y, por ejemplo, la legislación vigente en las administraciones públicas españolas, establece la valoración y por lo tanto los calendarios de disposición sobre las series documentales. Esto, traducido a un EDRMS, obligaría a la implementación a través de los módulos de RM que mantienen el *file plan*, haciendo mucho más difícil la implementación sobre sitios o tipos de contenidos u objetos tal como proponen los dos ejemplos explicados más arriba.

Web recomendada

En la siguiente dirección se puede ver la descripción de las nuevas funcionalidades de Sharepoint 2013, incluyendo las políticas de retención a nivel de sitio.

<http://technet.microsoft.com/en-us/sharepoint/fp142374.aspx>

Web recomendada

Se puede consultar en la siguiente dirección el folleto comercial con la descripción de esta funcionalidad.

<http://www.emc.com/collateral/software/data-sheet/h3130-rps-ds.pdf>

En ambos casos, es necesario tener presente desde el principio de la implementación cómo se aplican estas políticas en la organización, puesto que esto nos puede determinar cómo definir el modelo de información (estudiado en el apartado 2.3 de este módulo).

En las herramientas EDRMS las políticas de disposición son una implementación predefinida de un conjunto de los *workflows* y reglas que hemos estudiado en el apartado 2.5. Habiendo entendido el funcionamiento general de estos objetos, será fácil entender cómo se definen y aplican las políticas de retención y disposición.

El objetivo general de implantación de las políticas de retención y disposición en un EDRMS es la automatización de los procesos de ejecución de las mismas, especialmente las acciones de destrucción de documentos una vez que se han cumplido sus plazos de vigencia.

6.2. Conceptos clave y su implementación

En la mayoría de los EDRMS la implementación de las políticas de disposición se hace mediante la personalización de unos objetos predefinidos, que se definen de forma independiente para luego asociarlos con una determinada clase o categoría del *file plan*.

Para poder implementar las políticas de disposición es importante entender bien cómo se relacionan estos objetos. En la explicación de este tipo de objetos vamos a utilizar la terminología en inglés, ya que a pesar de todos los esfuerzos normalizadores de la traducción, las distintas herramientas lo han hecho de diferentes formas, lo que dificulta que podamos hablar en general de las mismas. En inglés hay una mayor uniformidad en los temas utilizados, que además están directamente asociados con los términos de uso común en los manuales y estándares de la gestión documental.

6.2.1. “Disposition schedules”

El objeto principal que hay que definir es generalmente reconocido como *disposition schedule*.

Los *disposition schedules* son habitualmente una combinación de eventos predefinidos, que se componen de los siguientes elementos:

- Un **detonador** o *trigger*, que indica que debe comenzar el periodo de retención. Normalmente, los tipos de detonadores posibles están predefinidos y se basan en los atributos de fechas asociados a los documentos.
- El **periodo de retención** o calendario de disposición, definido normalmente en años, pero que también puede admitir unidades más pequeñas de tiempo. A los tiempos definidos de forma global, algunas herramientas le añaden un periodo de gracia, expresado normalmente en forma de días,

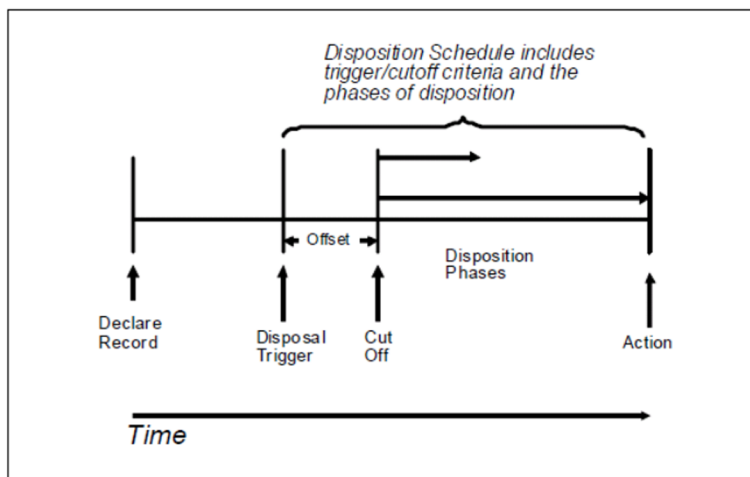
Ved también

Los eventos predefinidos los hemos estudiado en el apartado 2.5, “Los tipos o clase documentales: cómo implementar los metadatos”.

que muchas herramientas han implementado con *cut off*. No tiene más propósito que añadir unos días de “revisión” antes de que empiece a contar “oficialmente” el tiempo del calendario.

- Finalmente, incluye las **acciones** a las que va a dar lugar. Estas acciones también están predefinidas habitualmente y se refieren a la destrucción o transferencia a otro emplazamiento. Las acciones se pueden encadenar definiendo las posibles diferentes etapas de la disposición.

Figura 18. Visión de la definición de la disposición en IBM FileNet RM



Fuente: Understanding IBM FileNet Records Manager

La base del funcionamiento del mecanismo de la disposición es la definición del *trigger* o detonador. En las herramientas más sofisticadas se pueden definir nuevos eventos o *triggers* a partir de los cuales se comienza a contar el periodo de retención. Pueden estar basados en fechas o datos, pero lo importante para la implementación es que esa fecha o dato sea uno de los metadatos que se han capturado junto con el documento.

Por ejemplo, si se ha definido un calendario asignado a la acción finalización de la auditoría, necesitamos obligatoriamente que la fecha finalización de la auditoría sea un metadato capturado al nivel de agrupación al que asignar el calendario.

Esta necesidad nos indica lo importante que es tener en cuenta la disposición en el momento de la definición del modelo de información (estudiado en el apartado 2.3).

En otros módulos RM, como por ejemplo Collabware for Sharepoint, la disposición se ha incluido en un concepto más amplio denominado *compliance policy*, en el que se asocia la forma como los documentos se declaran records con los distintos conceptos habitualmente asociados a un “Disposition Schedule”.

Lo más importante para la implementación es entender que se debe hacer una abstracción de las posibilidades de *disposition schedules*. Lo habitual es que si la organización ha evaluado o valorado los documentos, tengamos distintos comportamientos posibles pero siempre asociados a un determinado tipo de

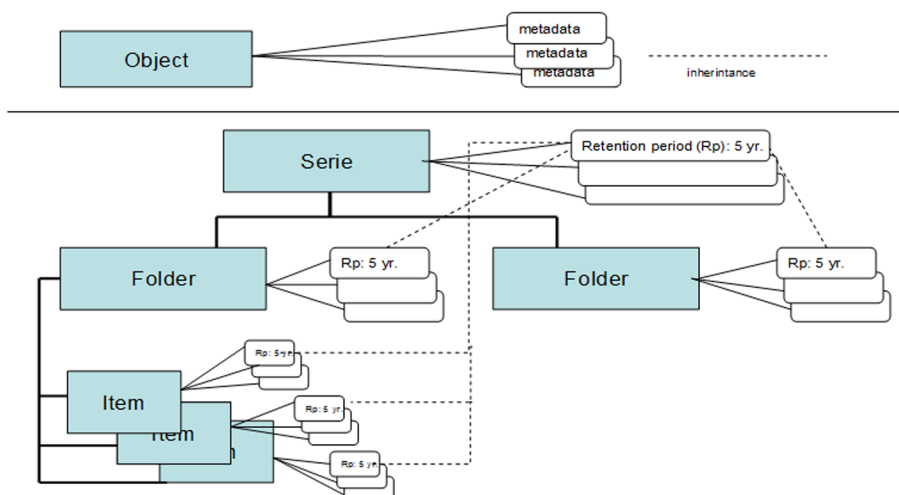
documento o serie. La primera labor será definir comportamientos comunes que nos permitan reutilizar los *disposition schedules* para distintos tipos de documentos o series.

6.2.2. Asignación al File Plan

La segunda parte de la implementación de la disposición es la asignación de los *disposition schedules* a la categoría correspondiente del *file plan*. Para poder hacer esta asignación sin problemas, es necesario que el *file plan* se haya definido de forma que sea posible aplicarlas sin el establecimiento de excepciones.

También es necesario entender que cuando se están asignado los *disposition schedules* a la clasificación o *file plan*, lo que estamos haciendo es asignar los metadatos correspondientes a nivel de documento. En base a la herencia de metadatos, las acciones previstas aplicadas a un nivel de la clasificación son heredadas por todos y cada uno de los documentos, sin tener que haber realizado una asignación de metadatos individualizada.

Figura 19. Ejemplo de herencia de metadatos sobre disposición



Fuente: ISO 23081-2

Nota

En el esquema de metadatos de las NTI e-EMGDE, existen toda una serie de elementos relacionados con la disposición, que en el caso de los EDRMS (especialmente los que cumplan con MOREQ2010) serán proporcionados por funcionalidades ya existentes y que, por lo tanto, no tendrán que hacerse explícitos en los metadatos de usuarios ni cumplimentarlos cada vez que se captura un documento.

Cuando asignamos los *disposition schedules* a las distintas categorías de *file plan*, es habitual que se nos pida informar sobre la *disposition authority*. Los ejemplos que habitualmente usan los manuales al uso suelen despistar bastante sobre su significado y cómo utilizarlo. Lo que se pretende en este punto es simplemente documentar en qué se apoya la asignación de este calendario o *disposition schedule* a esa agrupación documental. Esto puede ser una referencia a una ley o disposición legal específica (sería el caso de las administraciones públicas españolas que publican sus decisiones de valoración en los res-

pectivos boletines oficiales) o hacer referencia a un procedimiento interno. En muchos EDRMS, para facilitar la implementación se proveen de herramientas para definir las *disposition authorities* de forma que pueden ser reutilizadas, ya que normalmente se aplican en más de un caso.

Los EDRMS con funcionalidades más sofisticadas nos pueden permitir programar excepciones a la aplicación de los *disposition schedules* asignados a un *file plan*; permitiendo que estos actúen de distinta forma sobre los documentos agrupados en una misma categoría. Estas excepciones pueden plantarse basándose en el tipo de objeto o contenido, o en los valores de ciertos atributos. En todos los casos, utilizar este tipo de funcionalidades requiere de un estudio en profundidad, ya que su complejidad no los hace aconsejables en todos los casos. Cuando esto se aplica, lo que estamos es rompiendo la herencia de metadatos de disposición y aplicándola a nivel de cada documento en particular.

6.2.3. Los procesos de disposición

Una vez que se ha asignado el calendario de disposición o *disposition schedule*, el proceso automatizado es definido utilizando las mismas herramientas y criterios de cualquier otra automatización de procesos. Se puede determinar que algunas acciones se ejecuten automáticamente a requerimiento de una persona con los permisos necesarios o que requiera de una o varias aprobaciones.

Cuando un *disposition schedule* define una acción de destrucción o transferencia de documentos electrónicos, normalmente los EDRMS no solo se limitan a ejecutarlas, sino que nos ofrecen funcionalidades que nos permiten documentar la acción.

Tanto en los procesos de destrucción como de transferencia puede existir una funcionalidad para crear un registro o lista de los documentos que han sido destruidos o transferidos.

De la misma manera que en la automatización de procesos se pueden programar condiciones excepcionales o excepciones sobre el proceso general, también pueden programarse excepciones sobre la ejecución de la disposición. En este caso es necesario que de forma deliberada se marque mediante un metadato que a una determinada agrupación documental o documento no le va aplicar la regla y calendario de disposición asignado. Esta excepción (en inglés, *hold*) se supone que se aplica a circunstancias excepcionales que suspenden el cómputo de plazos, como por ejemplo que el asunto sobre el que versan los documentos esté sometido a un proceso en los tribunales.

Los distintos EDRMS ofrecen esta funcionalidad en distintas modalidades, que comparten la filosofía general: de forma deliberada y “manual” se excluyen algunos documentos del proceso que les corresponde de disposición.

Normalmente los módulos RM dejan en manos de un rol específico las funcionalidades que permiten la personalización de la disposición. Este rol específico está predefinido para los especialistas en gestión de documentos.

6.3. La preservación en los EDRMS

Todos los proyectos de gestión documental que incluyan documentos que tengan que conservarse a largo plazo deben tener una visión de futuro e incluir en su planificación los aspectos relacionados con la preservación digital. Sin embargo, los EDRMS no incorporan habitualmente herramientas que permitan ejecutar las acciones de preservación.

Como hemos visto en apartados anteriores, se podrían implementar acciones de conversión de formatos asociadas a reglas, eventos o *workflows*, pero no como base de un plan de preservación planificado.

Lo mismo sucede con las funcionalidades referidas al mantenimiento de las características de objetos específicos, como pueden ser las firmas.

Hoy por hoy, cualquier funcionalidad relacionada con la preservación que se quiera incluir en un EDRMS deberá ser programada y definida de forma específica.

7. Búsquedas y visualizadores

En este apartado estudiamos las funcionalidades que una plataforma de gestión documental puede ofrecer para la búsqueda y visualización de la información y los documentos almacenados en ella.

7.1. Aspectos relacionados con la búsqueda

Todos los EDRMS incluyen potentes motores de búsqueda que permiten recuperar los distintos objetos o contenidos que gestionan. Cómo utilizar las posibilidades que nos ofrecen estos motores de búsqueda forma parte del proceso de implementación de la herramienta.

Normalmente todos los EDRMS nos proponen algún tipo de interfaz de usuario para interactuar con el repositorio (veremos las distintas posibilidades de interfaces en el siguiente apartado). Todas estas interfaces, sean del tipo que sean, nos van a proponer una forma de búsqueda predefinida, basada en el modelo de información estándar. Para ajustarla a nuestro propio modelo definido, siempre será necesaria una personalización en la implementación.

Las interfaces también nos van a ofrecer habitualmente un tipo de vista determinada sobre los contenidos del repositorio. Esta se presenta en forma jerarquizada de carpetas y documentos sobre la que se nos permite navegar. Esta vista en realidad no es más que el resultado de la ejecución de una búsqueda determinada.

La personalización de la búsqueda será necesariamente mayor cuanto más personalizado sea el modelo de información definido y los tipos de objetos personalizados; aunque siempre es necesario un mínimo.

En todos los EDRMS vamos a encontrarnos la posibilidad de dos tipos de búsqueda:

- La búsqueda por tipos de objetos y los distintos atributos o metadatos.
- La búsqueda por contenido de aquellos objetos con contenidos textuales.

7.1.1. La búsqueda por atributos

La búsqueda por atributos o metadatos está obligatoriamente relacionada con la definición de los tipos de objetos y los atributos o propiedades definidas.

En principio los motores de búsqueda asociados a los EDRMS serán capaces de buscar sobre todos los tipos de contenidos utilizando los atributos comunes a todos ellos. Por ejemplo, es muy normal que se ofrezca una búsqueda simple que busca sobre el campo Title o Name que es común a todos los tipos de objetos.

Si se desea una búsqueda sobre otros atributos o campos específicos que se hayan creado al definir los tipos de objetos, lo habitual es que tengamos que definirla partiendo de dos posibilidades básicas:

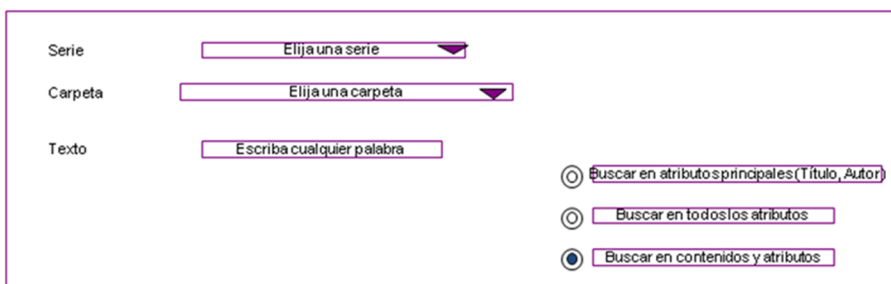
- definir cuáles son los atributos sobre los que se quiere buscar;
- definir plantillas de búsqueda para los usuarios.

La forma más sencilla será obligar al usuario a elegir el tipo de contenido, y a partir de esta primera elección, proponerle en pantalla los atributos sobre los que puede buscar.

Sin embargo, si se han definido muchos tipos de objetos o si los nombres asignados no son intuitivos para el usuario, esta opción simple puede ser inviable. También puede serlo si se pretende buscar en más de un tipo de objeto al mismo tiempo, por ejemplo, carpeta o *folder* y documento o *document*.

En estos casos será necesario personalizar el *template* o plantilla de búsqueda, asociándolo a las distintas sentencias de búsqueda. En la figura 20 presentamos un diseño personalizado de plantilla de búsqueda en donde se obvian los distintos tipos de objetos.

Figura 20. Diseño personalizado de plantilla de búsqueda



El formulario muestra tres campos de selección:

- Serie:** Un menú desplegable con el texto "Elija una serie".
- Carpeta:** Un menú desplegable con el texto "Elija una carpeta".
- Texto:** Un campo de texto con el placeholder "Escriba cualquier palabra".

Debajo de estos campos hay tres opciones de búsqueda con botones de radio:

- Buscar en atributos principales (Título, Autor)
- Buscar en todos los atributos
- Buscar en contenidos y atributos

Es importante tener en cuenta cuáles son los atributos que han sido marcados como indexables. En algunos casos, cuando se aplican opciones avanzadas de búsqueda mediante herramientas de terceros o módulos específicos, puede ser necesario marcar específicamente los atributos buscables.

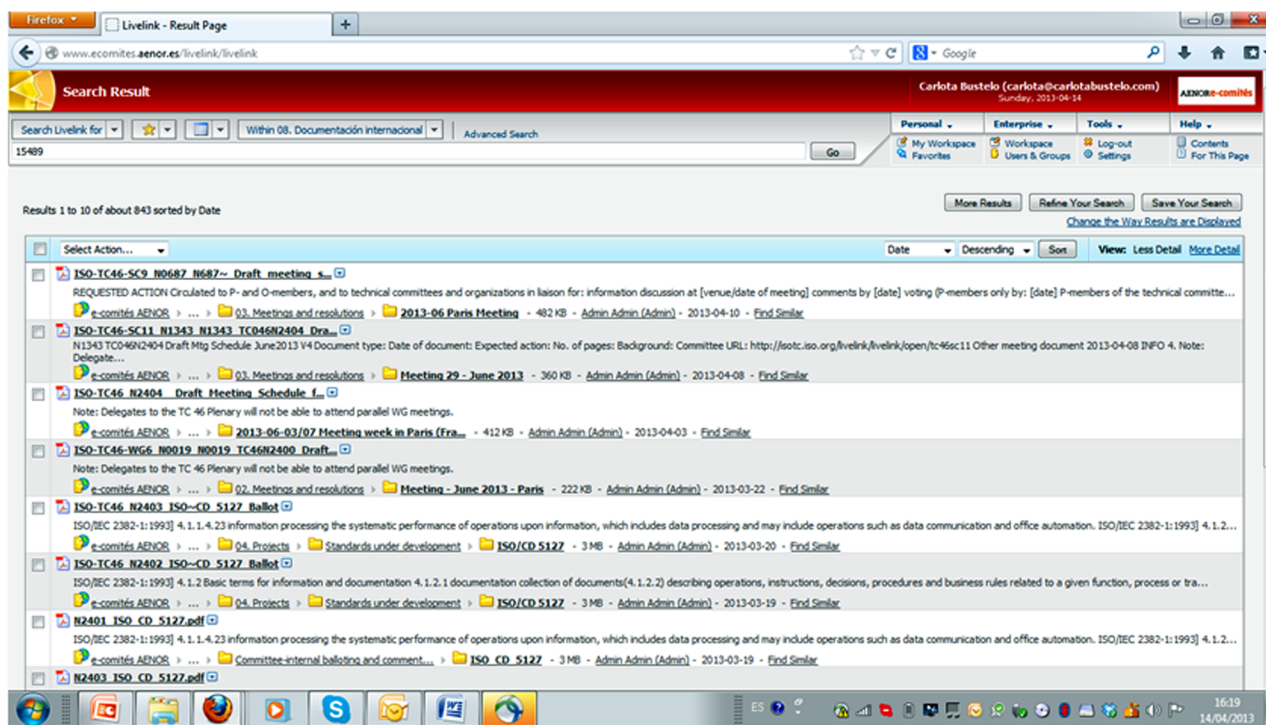
7.1.2. Búsqueda por contenidos

Todos los motores de búsqueda de los EDRMS permiten la búsqueda por contenidos textuales. En la implementación de este tipo de búsquedas hay que tener en cuenta los siguientes factores:

- Para que este tipo de búsquedas se produzcan y satisfagan las necesidades de los usuarios se requiere la construcción de potentes índices de búsqueda.
- Dependiendo del volumen de documentos a tratar, la creación y mantenimiento de estos índices puede requerir recursos de hardware y software dedicados, por lo cual esto debe estar planificado de antemano con los responsables de la arquitectura de sistemas.
- Es posible que se tenga que programar la indización con cierto retardo, lo cual puede causar frustración en los usuarios que saben que existe un determinado documento, pero no son capaces de recuperarlo por la búsqueda de contenidos.

Para la implementación de este tipo de búsqueda tendremos que decidir cómo presentarla al público con diferentes opciones. Vamos a estudiar cuáles son estas con el ejemplo de la figura 21, que es una captura de pantalla de la aplicación e-comites de ISO sobre la plataforma Livelink de Open Text.

Figura 21. Captura de pantalla de resultados de búsqueda en la aplicación e-comites sobre Livelink de Open Text



En la esquina superior izquierda nos muestra cómo se ha iniciado la búsqueda por contenido. En este caso, para evitar la búsqueda sobre todo el contenido del repositorio (hay que tener en cuenta que primero se ejecuta la búsqueda y luego se filtra por las ACL que permiten tener acceso a ella), nos permite iniciar la búsqueda desde un punto determinado de la jerarquía sobre la que podemos navegar en la vista principal. De esta manera se consigue acotar el conjunto de documentos donde realizar la búsqueda (en el ejemplo, los que están bajo la carpeta 08. Documentación internacional).

En la barra azul, encima de los resultados a la derecha, nos encontramos con las opciones de poder ordenar los resultados por distintos criterios. Esta opción es muy interesante implementarla pues habitualmente los resultados que se obtienen con este tipo de búsquedas no son muy precisos.

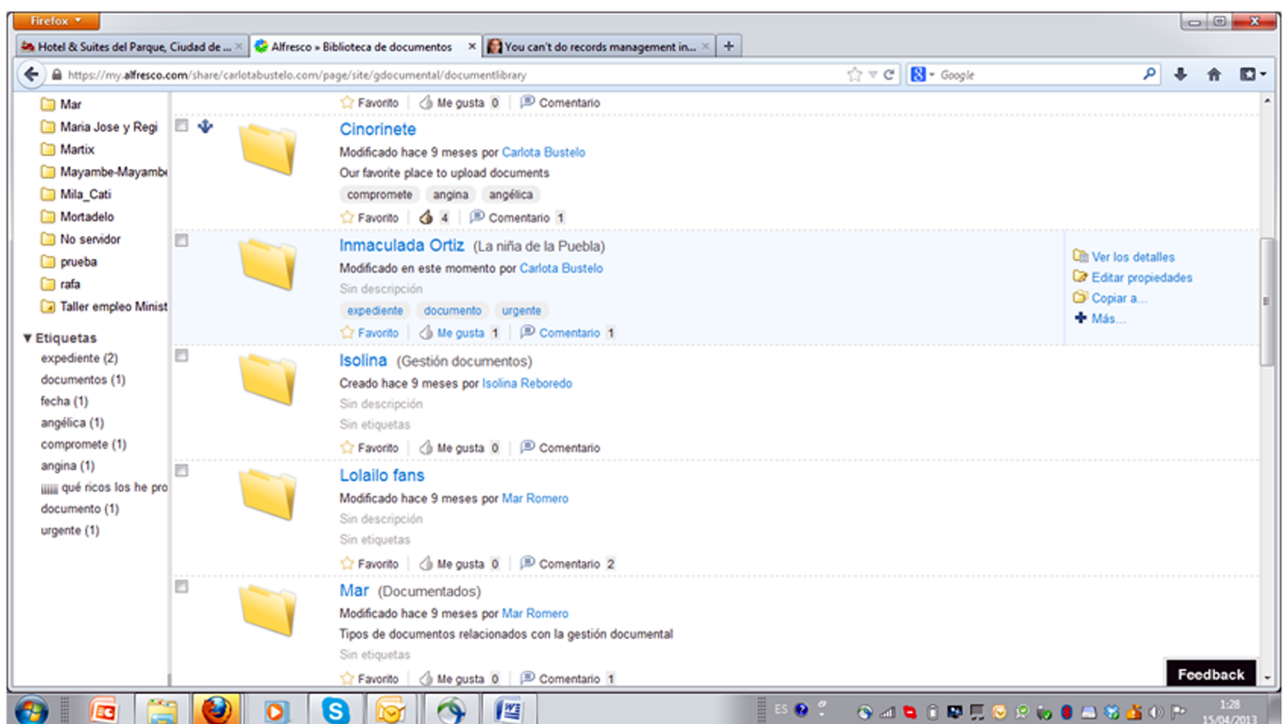
Los resultados de búsqueda nos muestran solo los tipos de objeto *document*, no las carpetas, aunque ha buscado también en los atributos de las mismas. En este ejemplo lo importante es cómo se ha definido la vista, enseñando no solo el nombre de los objetos recuperados, sino también la ruta de carpetas donde se encuentran, lo que nos permite entender mucho mejor de qué se trata antes de acceder al mismo.

7.1.3. Otras posibilidades

En cuanto a las búsquedas y sus resultados, deberemos tener en cuenta que en la implementación se puede plantear la realización de vistas específicas que presenten un determinado tipo de información en una visualización diferente a la que se ofrece por defecto. Esta decisión de implementación requerirá con probabilidad un desarrollo a medida o la realización de una aplicación, a no ser que podamos resolverlo con otras funcionalidades que nos ofrezcan algunas de las herramientas:

- La posibilidad de guardar búsquedas, para ejecutarlas cuando se necesiten.
- La posibilidad de buscar documentos parecidos o similares.
- Aumentar las posibilidades de búsqueda, permitiendo el etiquetado social por parte de los usuarios.

Figura 22. Etiquetado social en Alfresco. Captura de pantalla de Alfresco en la nube



7.2. Visualizadores de documentos

Algunos EDRMS ofrecen como parte de sus funcionalidades el uso de visores o visualizadores de documentos. También pueden encontrarse estas herramientas como productos terceros que pueden ser incorporadas a los mismos.

Los visores o visualizadores son piezas de software que nos permiten visualizar los documentos sin utilizar las aplicaciones nativas en las que han sido creados. Por ejemplo, seríamos capaces de visualizar un documento Word (con formato, docx) sin necesidad de tener instalado en el ordenador Microsoft Office.

Los visores son muy interesantes en la consulta de planos realizados con aplicaciones auto-cad por usuarios que no saben manejarlos.

Además de la ventaja importante de no necesitar la aplicación nativa, los visores son también utilizados en entornos distribuidos, con servidores lejanos y comunicaciones regulares, pues permiten consultar los documentos sin necesidad de descargarlos.

La utilización de visores permite además la inclusión de anotaciones y marcas de los documentos en capas que se superponen al documento original. En la figura 23 vemos cómo se presentan estas funcionalidades en el folleto comercial de Knowledge Lake Imaging, Visualizadores para Sharepoint.

Con esta misma tecnología, es posible ofrecer funcionalidades de *redaction* enmascaramiento o anonimización (que hemos visto en el apartado 2.4).

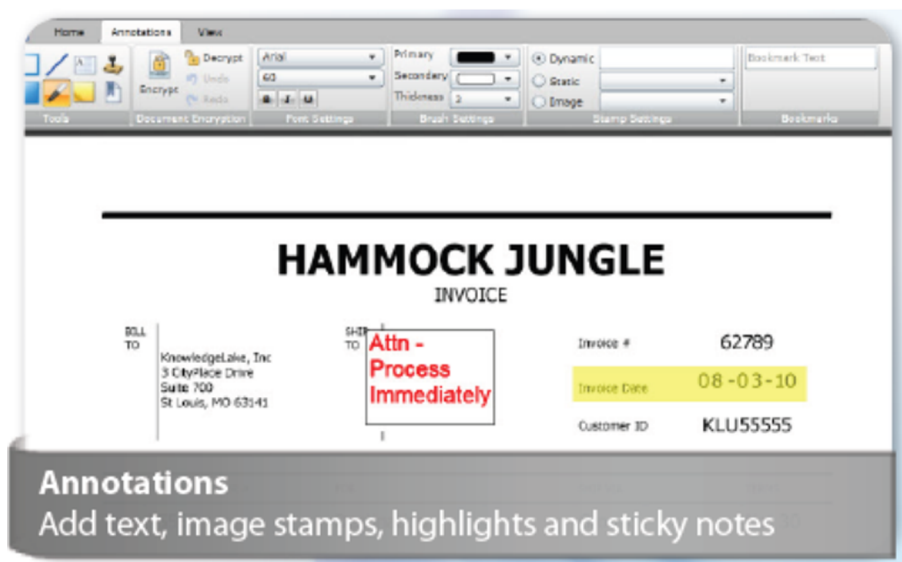
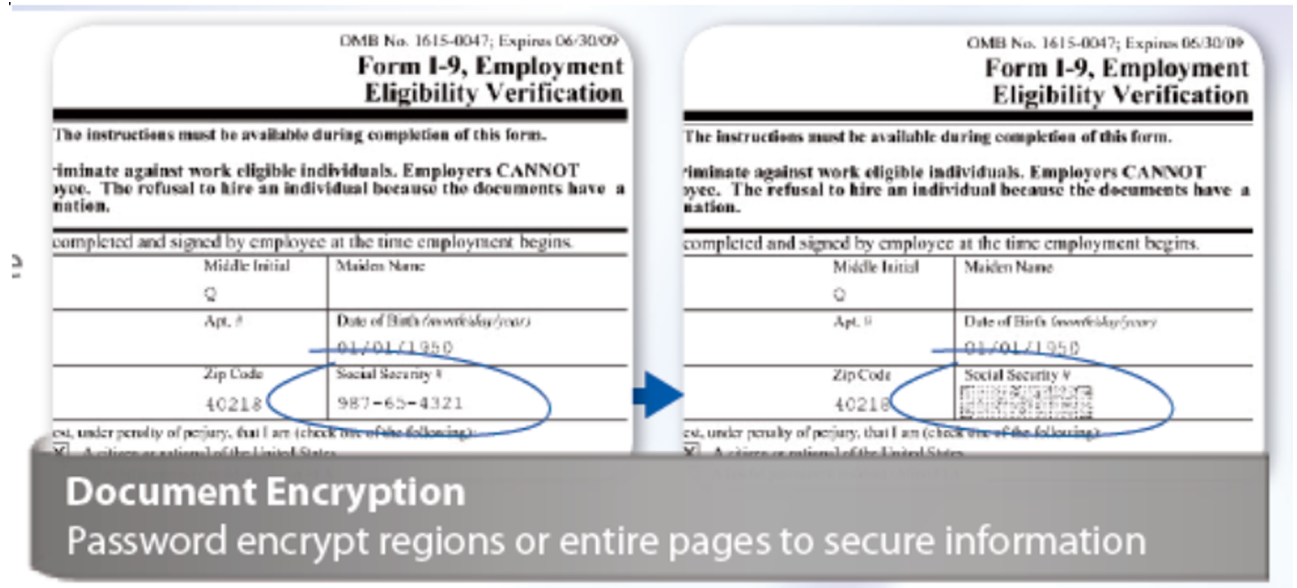


Figura 23. Anotaciones y encriptación en un visor



Fuente: Folleto comercial de Knowledge Lake Imaging

Cuando utilizamos estos visores, a la hora de implementar debemos definir en qué momento de la consulta del documento se van a activar; si es “la vista por defecto” de todos los documentos, solo de algunos o es una funcionalidad que se activa en un momento determinado o para unos usuarios determinados.

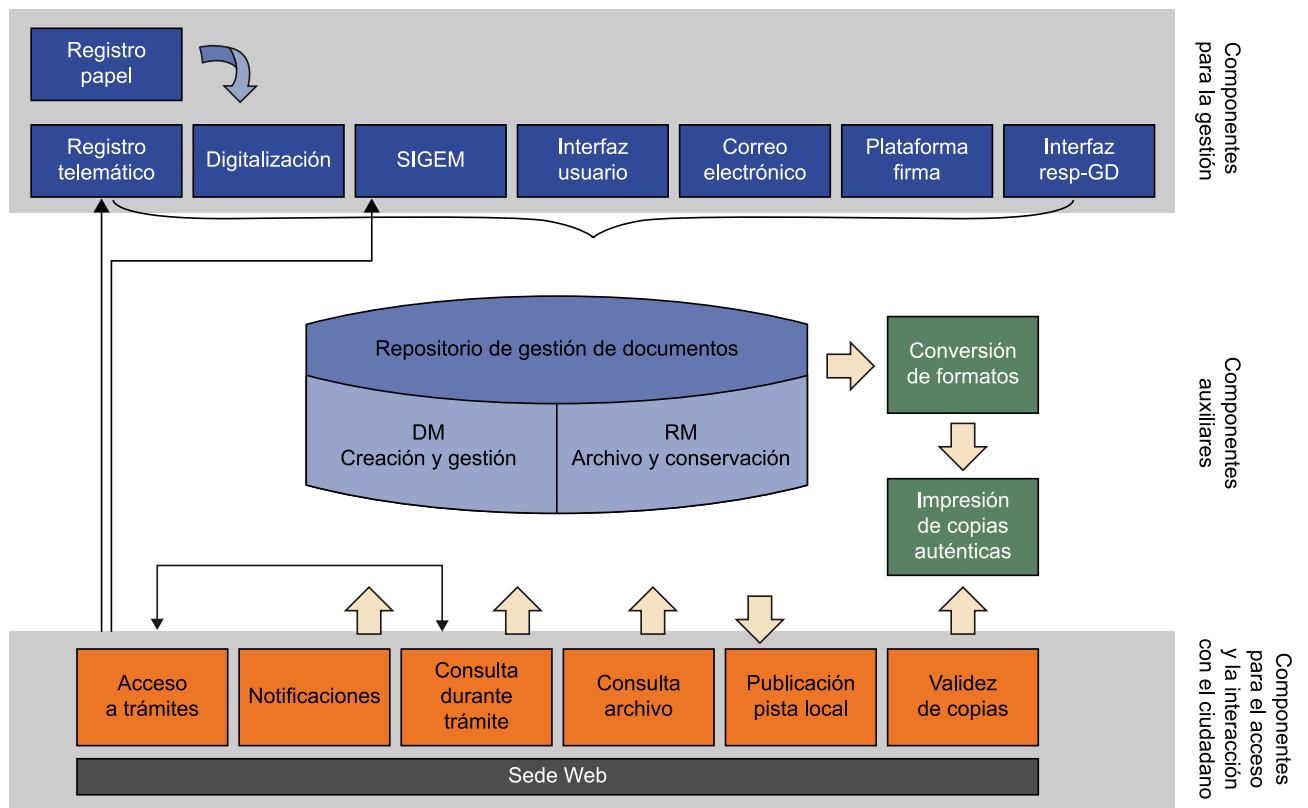
8. Integración con otras aplicaciones

Los proyectos de implementación de un EDRMS se desarrollan frecuentemente en entornos en los que es necesario integrar diferentes aplicaciones de negocio, que interactúan con el repositorio de gestión de los documentos electrónicos.

En este apartado estudiaremos las tres posibilidades de integración existente: la integración predefinida por las propias herramientas, que se ofertan en el mercado de forma integrada; la integración de aplicaciones de gestión existentes en la organización; y por último, vemos cómo se plantea la integración con aplicaciones específicamente diseñadas para la puesta en marcha del modelo de gestión documental.

Tener definida esta integración es una condición necesaria para el éxito de la implantación. Los documentos electrónicos están normalmente relacionados con muchas otras aplicaciones que ya existen en la organización o que son desarrolladas específicamente para el proyecto de gestión de documentos electrónicos.

Figura 24. Componentes del modelo de gestión de documentos electrónicos en una administración pública española



En la figura 24 nos encontramos un diagrama de las distintas piezas a encajar en un proyecto real en una administración pública española. Cada una de las piezas es una aplicación diferente.

8.1. Integración predefinida

En este apartado haremos un repaso de los tipos de integración que habitualmente se incluyen en los propios EDRMS, bien como parte del paquete de software, bien como piezas independientes que pueden ser añadidas según las necesidades. Repasaremos las interfaces de acceso directo a las aplicaciones, la integración del correo electrónico y otras herramientas ofimáticas, la integración como aplicaciones de digitalización y repasaremos el concepto de “conectores”.

8.1.1. Interfaces de usuario

Los EDRMS ofrecen habitualmente unas interfaces o varias para que el usuario final sea capaz de interactuar con el repositorio de gestión documental. Lo normal es que se ofrezca al menos un cliente web, que no necesita de ningún tipo de instalación en los ordenadores cliente.

El cliente web se acompaña muchas veces de otros tipos de clientes que se instalan en los ordenadores desde donde se va a acceder a los repositorios y que pueden integrarse en el explorador de Windows o simplemente ofrecer otro tipo de acceso al repositorio.

Estos clientes presentan en algunos casos posibilidades de personalización, que en muchas organizaciones se consideran insuficientes.

La decisión de utilizar o no los tipos de clientes que nos ofrece la herramienta es otra de las decisiones de implementación y las variedades son muchas. Desde organizaciones que determinan que no hay ninguna necesidad de estas interfaces de usuario, pues siempre se interactuará con los documentos desde las aplicaciones de gestión, hasta organizaciones que deciden implementar una combinación de posibles interfaces de usuario, hasta la que decide desarrollar y programar su propia interfaz de usuario.

Cuando se implementan los módulos RM, es más que frecuente que parte de los mismos sea una interfaz de usuario específica para los responsables de gestión documental, que es implementada de forma diferente al resto.

8.1.2. Aplicaciones ofimáticas

Muchos EDRMS ofrecerán funcionalidades para facilitar la integración de las herramientas más comunes en las organizaciones con el repositorio.

Estas herramientas se ofrecen normalmente sobre la *suite* de productos de Microsoft Office y permiten guardar documentos directamente en el repositorio sin haber salido de la aplicación.

Cuando se aplica sobre MS Word, por ejemplo, el resultado suele ser que en la barra de herramientas propia del Word aparece una indicación del propio EDRMS que permite al menos guardar el documento directamente en el repositorio sin haberlo guardado en el disco duro.

Estas funcionalidades que despiertan gran interés cuando se explican, suelen tener menos uso del esperado cuando se implementan, pues los usuarios rechazan fácilmente cambios en las herramientas de uso cotidiano que ya conocen a la perfección. Esto sucede con la excepción de la integración con Outlook para la gestión del correo electrónico.

La posibilidad de “documentar” algunos correos electrónicos conjuntamente con otros documentos es en muchos casos un fuerte requisito de las organizaciones con respecto a la gestión documental.

La implementación de esta integración requiere del estudio de la utilización de los tipos de objetos o contenidos que se van a utilizar para los correos electrónicos.

Vídeo recomendado

Para el entendimiento de cómo funcionaría la integración del correo electrónico, es ilustrativo el vídeo sobre esta funcionalidad en Documentum. Es un vídeo sin voz que nos muestra cómo desde Outlook vemos las carpetas o *folders* que tenemos definidas en el repositorio, pudiendo arrastrar un correo recibido.

http://www.youtube.com/watch?v=OXC7HTm_dsE

8.1.3. Aplicaciones de digitalización

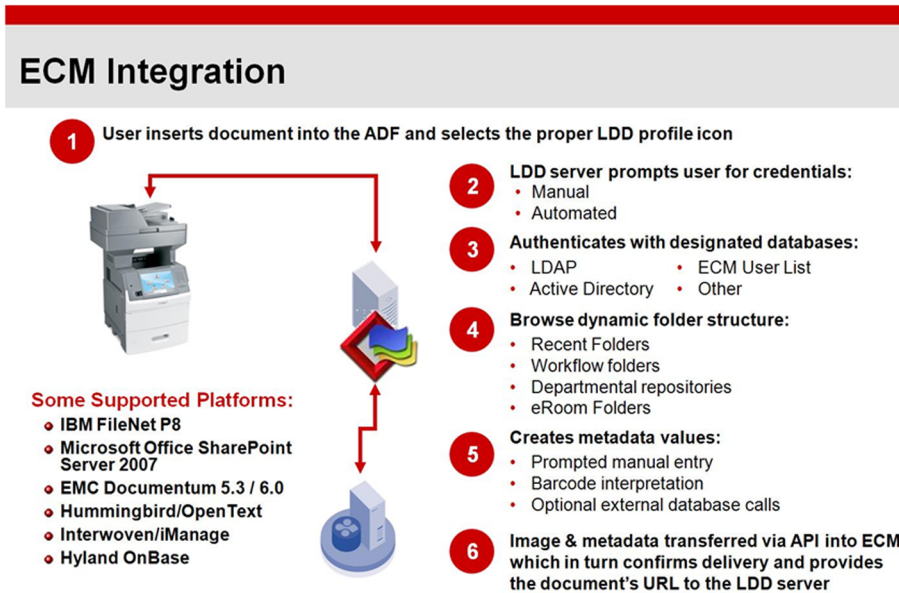
Otro de los módulos típicos de integración predefinidos en los EDRMS es el módulo de captura o digitalización.

Normalmente las aplicaciones de digitalización se comercializan de forma separada, pero con una integración muy definida con cada uno de los EDRMS existentes en el mercado. Por ejemplo, Documentum incluye en su comercialización la integración con los documentos de Captiva (comprado por EMC). Por su parte, Filenet comercializa la integración con los productos Kofax.

Cuando implementamos productos que ya tienen estas integraciones, lo normal es que nos encontremos con una gran facilidad para integrar en el repositorio correspondiente los documentos escaneados o digitalizados. Esto quiere decir, por ejemplo, que los documentos escaneados se integran directamente en el repositorio permitiendo la elección del tipo de objeto o habiéndola definido de antemano.

Cuando, por el contrario, implementamos soluciones de digitalización externas tendremos que predefinir cómo será la interacción con el repositorio. En la siguiente figura 25 vemos el esquema de integración que propone Lexmark, uno de los proveedores de soluciones de digitalización que se aplican con diferentes EDRMS.

Figura 25. Integración digitalización ECM



Fuente: Folleto explicativo soluciones Lexmark

8.1.4. Conectores

Algunos EDRMS ofrecen conectores con aplicaciones específicas de gran implantación. Uno de los más comunes es la integración con SAP (el ERP³ más extendido en el mercado).

⁽³⁾ERP - Enterprise Resource Planning.

Cuando la implementación pasa por el uso de alguno de estos conectores, hay que estudiar detalladamente el uso de los mismos y cómo implementarlos.

Ejemplo

Como ejemplo de todas las distintas facetas que hemos visto en este apartado de integraciones predefinidas, se puede consultar la página de lo que ofrece la herramienta Ever-suite a este respecto:

<http://www.ever-team.com/es/productos-ecm-funcionales/conectores-ecm.html>

8.2. Integración de aplicaciones existentes

Otro de los escenarios posibles es la necesidad de integración de los EDRMS con aplicaciones ya existentes en la organización.

Para hacer esta integración hay que contar sin duda con expertos informáticos que sean capaces de personalizar y programar cómo se hace la integración. Sin embargo, es importante entender los fundamentos de esta integración para poder aportar.

Lo habitual es que esta integración se haga actualmente siguiendo el principio de la arquitectura abierta (*service-oriented architecture, SOA*). Algunas de las herramientas del mercado dicen ofrecer una serie de herramientas o servicios (*web services*) que las aplicaciones podrían utilizar para interactuar con el repositorio, pero también es habitual que por parte de los departamentos de tecnologías de información se decida desarrollar un bus de servicios (*ESB- enterprise services bus*) totalmente independiente. La idea central es tener una capa intermedia completamente independiente de las aplicaciones. Cada servicio web recibe una petición de una aplicación y la transforma para acceder al EDRS, y viceversa.

Prácticamente se necesitaría un servicio para cada una de las funcionalidades básicas que provee la plataforma (por ejemplo, buscar documentos o realizar el *check out*). Los servicios deben ser “expuestos” para que cualquier aplicación pueda utilizarlos o invocarlos. Esto significa que las aplicaciones que se deben integrar deben ser modificadas para que sean capaces de utilizar estos servicios.

Figura 26 Enterprise Bus Service



Fuente: Web TCP Integración con ESB http://www.tcps.com/vermas/integracion_esb.htm

Teóricamente, aislar esta capa de servicios supondría la posibilidad de cambiar de aplicaciones o incluso de EDRMS y que los servicios siguieran funcionando. Pero para ello es necesaria una normalización de las funcionalidades ofrecidas como la que se intenta a través del protocolo CMIS⁴.

8.3. Creación de aplicaciones propias

También es frecuente encontrar implementaciones de un EDRMS en las que se haya decidido la creación de aplicaciones propias.

Para esto los proveedores pueden disponer de una serie de herramientas conocidas como API (*application programming interface*).

⁽⁴⁾Content Management Interoperability Services: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cmis

API = Conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

La construcción de aplicaciones propias utilizando las API o los *web services* de una determinada herramienta o plataforma ECM es una decisión a la que se debe llegar después de haber estudiado las necesidades específicas de la organización y haber determinado que realmente no están cubiertas con las herramientas y funcionalidades que nos ofrecen.

Una de las opciones que más se ha utilizado en las implementaciones de EDRMS es la programación de interfaces propias, que se ajustan perfectamente a las necesidades y características de la organización y de los usuarios que van a utilizarla.

Bibliografía

SharePoint Governance. Podéis leer en la web de AIIM lo que se considera los puntos clave de esta gobernanza: <http://www.aiim.org/Community/Wiki/view/Quick-Guide-to-SharePoint-Governance>

Sharepoint (2013). Políticas de retención a nivel de sitio: <http://technet.microsoft.com/en-us/sharepoint/fp142374.aspx>

Protocolo CMIS (*content management interoperability services*): https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cmis

