

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 1

Departamento de Técnicas de Programación

**Algoritmos de programación para la visualización y exportación
del repositorio del ECM Alfresco Community 3.0 desde el
cliente web Alfresco Explorer**

**Trabajo final presentado en opción al título de
Máster en Informática Aplicada**

Autor: Ing. Alexander Hernández Chapman

Tutora: Dra. C. Marely Del Rosario Cruz Felipe

Cotutora: MSc. Dunia Suárez Ferreiro

La Habana, octubre de 2013

Agradecimientos

A mis tutoras Marelys y Dunia.

A mis padres.

A mi hermano y su esposa Yami, por hacerme la vida más fácil y darme a la Luna (mi súper sobrina).

A mi familia, en especial mi prima Kelma.

A mis compañeros del proyecto Michel, Reinier, Pedro y Misael que sin ellos Alfresco sólo fuera un nombre.

A mis compañeros de trabajo Daynel, William, Yeneit y Delly.

A mis amigos de la vida Violena, Michel, Liván y su esposa Evelyn (por su inglés), Minerva, Héctor, Magalys y Virtudes.

A todos los que de una forma u otra hicieron posible la realización de este trabajo. GRACIAS!!!!!!!!!!!!!!!!!!!!!!

Dedicatoria

A mis padres.

Declaración jurada de autoría

Declaro por este medio que yo Alexander Hernández Chapman, con carné de identidad 78011307489, soy el autor principal del trabajo final de maestría “Algoritmos de programación para la visualización y exportación del repositorio del ECM Alfresco Community 3.0 desde el cliente web Alfresco Explorer”, desarrollada como parte de la Maestría en Informática Aplicada y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en La Habana a los ____ días del mes de _____ del año _____.

Resumen

Debido al gran volumen de información que se gestiona con el Gestor de Contenido Empresarial Alfresco Community 3.0 se hace necesario contar con herramientas capaces de realizar el respaldo o copia de seguridad a su contenido y de restaurarlo en caso de fallas de cualquier índole. Alfresco Community 3.0 no tiene incorporada ninguna funcionalidad capaz de realizar dichas acciones por lo que hay que recurrir al uso de herramientas externas, lo que provoca ineficiencias con respecto al tiempo de ejecución y al espacio que ocupan las copias de seguridad en el dispositivo de almacenamiento.

En la presente investigación se proponen algoritmos de programación, implementados en Java, para visualizar y exportar el repositorio del Alfresco Community 3.0 que al ser incorporados al cliente web Alfresco Explorer, mediante un módulo de paquetes de Alfresco, permiten gestionar las copias de seguridad logrando mejorar la eficiencia y usabilidad en las mismas.

Palabras clave

Alfresco Explorer, Copias de seguridad, Gestor de Contenido Empresarial Alfresco Community 3.0, Java.

Abstract

Due to the large volume of information that is managed with Enterprise Content Management Alfresco community 3.0 it needs tools that perform the backup or restore of the contents in case of failure of any kind. Alfresco Community 3.0 does not have any functionality built capable of performing such actions so we must resort to the use of external tools which cause inefficient runtime backups as well as the space that they occupy in the storage device when they save it.

In the present investigation are proposed programming algorithms, implemented in Java, to visualize and export the repository of Alfresco Community 3.0 that when incorporated into the Alfresco Explorer web client through an Alfresco module packages, allow you to manage backups achieving improved efficiency and usability.

Keywords

Alfresco Explorer, Backup, Enterprise Content Management Alfresco Community 3.0, Java.

Índice

Introducción.....	1
1. Copias de seguridad en los sistemas de archivos y sistemas de gestión de contenido empresarial	8
1.1. Gestión de contenidos empresarial.....	8
1.2. Sistemas de gestión de contenido empresarial	10
1.3. Análisis de las copias de seguridad en los ECM y los algoritmos utilizados para su ejecución.....	11
1.3.1. Copias de seguridad en los sistemas de gestión de contenido empresarial	12
1.3.2. Algoritmos para visualizar, seleccionar y exportar, desde un cliente web, los contenidos del repositorio	19
1.4. Algoritmos para recorrer estructuras de tipo árbol.....	21
1.5. Herramientas y tecnologías	23
Conclusiones parciales.....	25
2. Propuesta de solución para la Gestión de Copias de Seguridad en el ECM Alfresco Community 3.0	26
2.1. Algoritmo propuesto para la visualización del contenido	26
2.2. Algoritmos propuestos para exportar el contenido	29
2.3. Creación del AMP para integrar los algoritmos de visualización y exportación del repositorio del ECM Alfresco Community 3.0 desde el cliente web Alfresco Explorer	34
2.3.1. Integración de los algoritmos para la visualización y exportación en el AMP.....	35
2.3.2. Características de las funcionalidades del AMP propuesto.....	37
Conclusiones parciales	40
3. Valoración de la solución propuesta para la gestión de copias de seguridad desde cliente web Alfresco Explorer del ECM Alfresco Community 3.0.....	41
3.1. Valoración de la eficiencia en la gestión de las copias de seguridad de la solución propuesta.....	41
3.1.1. Análisis del espacio de almacenamiento.....	48

3.1.2. Análisis del tiempo de ejecución.....	51
3.2. Validación de la usabilidad	53
Conclusiones parciales.....	57
Conclusiones.....	58
Recomendaciones.....	59
Referencias Bibliográficas	60
Bibliografía	63
Anexos	65
Anexo 1: Encuesta Aplicada para medir la usabilidad.....	65
Anexo 2: Vistas del AMP propuesto.....	69
Anexo3: Artefactos de Ingeniería de Software generados en la etapa de diseño de la solución propuesta	77

Introducción

En la actualidad, la cantidad de información generada en distintas lenguas y formatos es abrumadora; tanto es así que de forma manual gestionar tales volúmenes de información resulta muy engorroso.

Gracias al surgimiento y desarrollo alcanzado por la gestión documental [1] existen formas de organizar y clasificar tales volúmenes de información para facilitar su gestión. Paralelo a ello y debido al desarrollo de las ciencias informáticas, se han implementado aplicaciones capaces de realizar la gestión documental a grandes volúmenes de documentos. Algunas de las aplicaciones utilizadas a nivel mundial para realizar la gestión de documentos son: Documentum [2], OpenKm [3] y el Gestor de Contenidos Empresarial (Enterprise Content Management, en inglés, abreviado ECM) Alfresco [4]. El uso de dichas aplicaciones brinda grandes ventajas debido a que proporcionan:

- Menor necesidad de trasladar los documentos.
- Mayor facilidad de recuperación y acceso a la información.
- Mejor uso de los recursos compartidos.
- Ahorro en gastos de copias y duplicados.
- Reducción de espacio de almacenamiento en locales de trabajos u oficinas.
- Mayor control en la gestión de los documentos.
- Mayor seguridad de los documentos.

El ECM Alfresco, en su versión Community 3.0, es el repositorio de contenido utilizado por el Gestor de Documentos Administrativos (GDA) eXcriba. El mismo consiste en una aplicación web desarrollada por el Departamento de Gestión Documental de la Universidad de las Ciencias Informáticas (UCI). Alfresco, como todo sistema de gestión de contenido, debe cumplir con las especificaciones que aparecen en el Modelo de Requisitos para la Gestión de Documentos Electrónicos de Archivos (MoReq).

MoReq rige las características que deben tener los Sistemas de Gestión de Documentos Electrónicos y Archivos (SGDEA) y en su especificación aparece textualmente: “(...)tanto la normativa legal como la propia actividad de la organización exigen que los SGDEA cuenten con procedimientos exhaustivos de creación periódica de copias de seguridad de los documentos de archivo y metadatos, de forma que sea

posible recuperar rápidamente cualquier documento de archivo perdido debido a un fallo del sistema, un accidente, un fallo en la seguridad (...) [5]. También se expone que los SGDEA deben permitirle al administrador programar rutinas de copias de seguridad donde este pueda: especificar su frecuencia y escoger las clases, expedientes o documentos de archivo a incluir en las mismas.

Además de los documentos que se gestionan, en los SGDEA existen archivos considerados esenciales, pues son absolutamente necesarios para el funcionamiento de dichos sistemas. Con ellos se pueden representar o modelar las funciones del negocio desde donde se encuentre ejecutándose el ECM Alfresco.

En el marco del Departamento de Gestión Documental, los archivos de configuración del ECM Alfresco son muy importantes, pues en ellos se encuentra la definición de los documentos o contenidos que en el GDA eXcriba se gestionan, por lo cual dichos archivos pueden ser clasificados como esenciales. MoReq recomienda para los archivos de configuración realizar una copia de seguridad de forma separada a los documentos incluidos en la copia.

El ECM Alfresco Community 3.0, en sus clientes web (Alfresco Explorer o Share) no propone ningún mecanismo que posibilite la realización de copias de seguridad fuera de su repositorio¹. Para realizar una copia de seguridad en el ECM Alfresco Community 3.0 se recomienda, en su página oficial [6], que se realicen una serie de acciones desde la consola del Sistema Operativo (SO). No obstante, la ejecución de las copias de seguridad desde la consola del SO tiene las siguientes limitantes:

- No cumplen con el requisito “Escoger los documentos de archivos a los que se hará copia de seguridad” especificado en MoReq, pues no se brinda la posibilidad en ningún SO, desde su consola, de escoger qué archivos o contenidos existentes en el repositorio del ECM Alfresco se desean incluir en la copia de seguridad. Por el contrario, obligatoriamente hay que almacenar todos los documentos existentes en el repositorio.

Lo anterior conlleva a la ineficiencia de la propuesta respecto al espacio utilizado en cualquier dispositivo de almacenamiento y el tiempo que demora su

¹ Dispositivo de almacenamiento externo (discos duros, memorias flash, etc.)

ejecución, para determinada cantidad de contenidos. De aquí la importancia de contar con un algoritmo que permita visualizar el repositorio de Alfresco y exportar sólo los contenidos seleccionados con el objetivo de disminuir el espacio y tiempo de ejecución de las copias de seguridad.

- No todos los usuarios o administradores que necesiten realizar una copia de seguridad tienen los conocimientos necesarios sobre SO y el Sistema Gestor de Bases de Datos (SGBD) para poder realizarlas a través de la consola de cualquier SO. A lo anterior se le suma la dificultad que representa querer ejecutar la copia de seguridad de forma periódica o programada, es decir, ejecutarla de forma automática cada cierto tiempo.

Para ello sería necesario sincronizar y realizar la copia de seguridad de la base de datos, de los índices de Lucene² y del sistema de archivos, todas ellas en el orden adecuado y con suficiente tiempo y sincronización entre ellas para no comenzar a realizar una sin haber terminado la otra. La falta de sincronización o una inversión en el orden de las copias provocarían la inconsistencia de los datos en la copia de seguridad creada.

- Debido a que en la ejecución de las copias de seguridad intervienen varias herramientas como son el SGBD (para realizar la copia de seguridad de la Base de Datos) o JConsole³ (para realizar la copia de seguridad de los índices de Lucene), se dificulta la recopilación de información sobre dichas copias, lo que constituye un inconveniente a la hora de realizar reportes sobre las mismas.

Alfresco Explorer es el cliente web que, por defecto, utiliza el Alfresco Community 3.0 y hasta la actualidad no se le ha incorporado ninguna funcionalidad para la exportación de copias de seguridad fuera del repositorio de Alfresco. El equipo de desarrollo y la comunidad mundial que contribuye con Alfresco ha centrado su trabajo en el Alfresco Share, aunque para ambos la manera en que se propone realizar las copias de seguridad en la página oficial de Alfresco es desde la consola del SO.

² Es una API para la recuperación o búsqueda de información de código abierto originalmente implementada en Java. Esta API provee a Alfresco de un motor de búsqueda para los metadatos y contenidos de los documentos que se encuentren en su repositorio [4].

³ Se utiliza para monitorizar el rendimiento y el consumo de recursos de las aplicaciones que se ejecutan en la plataforma de Java. Permite realizar el monitoreo de forma local o remota y genera gráficos teniendo en cuenta el consumo de memoria, de CPU, los hilos y las clases que son cargadas. Los parámetros pueden ser consultados y configurados a través de una Shell o de la interfaz gráfica que proporciona [7].

A partir de la situación problemática existente se identifica el siguiente **problema de investigación**:

¿Cómo mejorar la eficiencia y usabilidad en la gestión de copias de seguridad del ECM Alfresco Community 3.0?

Según [8] la eficiencia es la cantidad de recursos informáticos consumidos por un algoritmo. Considerando los recursos informáticos como tiempo de ejecución y espacio de almacenamiento que ocupan las copias de seguridad, en esta investigación se establece que mejorar la eficiencia consiste en disminuir el espacio de almacenamiento y el tiempo de ejecución en función de la cantidad de documentos.

De aquí que el **objeto de estudio** lo constituyan el proceso de ejecución de copias de seguridad en los sistemas de gestión de contenido empresarial y específicamente como **campo de acción** las copias de seguridad en el ECM Alfresco Community 3.0.

La investigación estará encaminada a cumplir con el siguiente **objetivo general**: desarrollar algoritmos de programación que permitan la visualización y exportación del repositorio del ECM Alfresco Community 3.0, desde el cliente web Alfresco Explorer, que mejoren la eficiencia y usabilidad en la gestión de las copias de seguridad.

Para el desarrollo de la investigación se trazaron los siguientes **objetivos específicos**:

1. Elaborar el marco teórico relativo a las copias de seguridad en los sistemas de gestión de contenido empresarial.
2. Implementar algoritmos de programación en Java que al ser integrados al ECM Alfresco Community 3.0, mediante un módulo de paquetes de Alfresco, permitan la visualización y exportación de su repositorio desde el cliente web Alfresco Explorer.
3. Valorar la eficiencia y usabilidad de la realización de las copias de seguridad desde el cliente web Alfresco Explorer, a partir de la incorporación al ECM Alfresco Community 3.0 de los algoritmos de programación propuestos.

Como vía para alcanzar los objetivos planteados, se definen las siguientes **tareas de investigación**:

1. Análisis crítico de las herramientas y tecnologías que permiten la gestión de copias de seguridad en los ECM open source.
2. Elaboración de la documentación y artefactos necesarios para el análisis, diseño e implementación de la solución propuesta.
3. Implementación de algoritmos que permitan la visualización y exportación del repositorio del Alfresco Community 3.0.
4. Integración de los algoritmos implementados al cliente web Alfresco Explorer ECM Alfresco Community 3.0.
5. Aplicación del cuestionario para medir la usabilidad de la solución propuesta.
6. Selección del método matemático para procesar los resultados del cuestionario aplicado.
7. Realización de la valoración a la solución propuesta.

Hipótesis de la investigación:

Con la implementación de algoritmos de programación que permitan la visualización y exportación del repositorio del ECM Alfresco Community 3.0, desde el cliente web Alfresco Explorer, mejorará la eficiencia y usabilidad en la gestión de sus copias de seguridad.

Entre los **métodos científicos** utilizados en esta investigación se destacan los siguientes:

Teóricos:

- Histórico-lógico: permitió consultar bibliografía referente al tema de investigación, toda su trayectoria, evolución y comportamiento.
- Análisis-síntesis: para establecer las definiciones vinculadas al tema de investigación, formular una propuesta a la situación que se estudia y la tecnología más apropiada para su desarrollo. Además para profundizar e investigar acerca de las partes que componen el objeto de estudio así como los procesos que intervienen en la realización de las copias de seguridad.

- **Sistémico:** se utilizó este método para estudiar la integración de las tecnologías utilizadas, mediante la determinación de sus componentes, así como la relación entre ellos. Esta relación determina por un lado la estructura y la jerarquía de cada componente y por otra parte su dinámica, siendo también la expresión del comportamiento del sistema como totalidad en que un componente depende de otro u otros.
- **Modelación:** para el desarrollo de los algoritmos propuestos que permiten la visualización y exportación del repositorio.

Empíricos:

- **Experimental:** para probar los algoritmos propuestos y así obtener una disminución del espacio y tiempo de ejecución.
- **Documental:** se utilizó para consultar bibliografía en fuentes de carácter documental tales como libros, artículos y ensayos.
- **Encuestas:** Con el objetivo de conocer el criterio de posibles usuarios finales con respecto a la usabilidad del módulo propuesto y la variante desde la consola del SO.

La principal contribución del presente trabajo es la propuesta de algoritmos, que integrados a un módulo para el cliente web Alfresco Explorer del ECM Alfresco Community 3.0 permiten al administrador:

1. Ejecutar y programar copias de seguridad en las que puede escoger los documentos o carpetas que serán incluidos en la copia de seguridad.
2. Realizar la copia de seguridad a los ficheros de configuración del ECM Alfresco.

El primero de los requisitos ofrece la posibilidad de disminuir, en dependencia del tamaño de los documentos incluidos, el tiempo de ejecución y espacio a ocupar por las copias de seguridad en cualquier dispositivo de almacenamiento logrando de esta manera una mejora de la eficiencia en su gestión.

El trabajo que se presenta tiene la siguiente estructura:

Introducción: define el diseño teórico y metodológico de la investigación.

Capítulo 1: expone los fundamentos teóricos acerca de los sistemas de gestión de contenidos empresariales y se realiza una valoración del estado del arte sobre la ejecución de copias de seguridad en los mismos. Se analizan los algoritmos utilizados para visualizar y exportar el repositorio desde sus clientes web. Además se presentan las tecnologías y herramientas para la implementación de la solución propuesta.

Capítulo 2: presenta la propuesta del conjunto de funcionalidades y algoritmos para la ejecución de copias de seguridad desde el cliente web Alfresco Explorer del ECM Alfresco Community 3.0.

Capítulo 3: describe la valoración sobre la solución propuesta para determinar si presenta mayor eficiencia y usabilidad que las variantes existentes desde la consola del SO.

Conclusiones: sección en la que se expone el cumplimiento satisfactorio de las tareas inicialmente planteadas.

Por último se encuentran las **recomendaciones**, la **bibliografía**, las **referencias bibliográficas** y los **anexos**.

1. Copias de seguridad en los sistemas de archivos y sistemas de gestión de contenido empresarial

En el presente capítulo se abordan aspectos teóricos sobre las copias de seguridad así como la gestión de contenido empresarial. Además se hace una caracterización de las herramientas que realizan copias de seguridad en sistemas de archivos y sistemas de gestión de contenido empresarial sobre la base de los algoritmos que utilizan para visualizar y exportar los contenidos.

Como primer paso es necesario realizar la definición de copia de seguridad o backup (en inglés, término ampliamente utilizado) para lo cual el presente trabajo toma la siguiente: copia de seguridad es la actividad de copiar, en algún dispositivo de almacenamiento, la información considerada importante existente en una computadora (archivos, bases de datos, fotos) para tenerla duplicada o conservada y así en caso de alguna falla (software o hardware) o pérdida de la información poder recuperarla. A la acción de recuperar la información se le denomina restauración [9].

1.1. Gestión de contenido empresarial

Antes de analizar cómo los ECM gestionan sus copias de seguridad se hace necesario definir algunos aspectos teóricos sobre la gestión de contenido empresarial y los sistemas que la realizan.

Según la Asociación para la Información y la Gestión de la Imagen (Association for Information and Image Management, en inglés, abreviado AIIM) la gestión de contenido empresarial son las estrategias, métodos y herramientas utilizadas para capturar, gestionar, entregar, almacenar y preservar el contenido y documentos relacionados con los procesos de la organización. Las herramientas y estrategias que propone la gestión de contenido empresarial permiten la gestión de la información no estructurada de una organización, siempre que esa información exista [10].

Los cinco términos claves que caracterizan a la gestión de contenido empresarial son: contenido, conformidad o cumplimiento, colaboración, costo y continuidad [10]. La continuidad es el único término que resulta de interés para la elaboración del presente trabajo, por lo que a continuación se describe brevemente:

La planificación de la continuidad del negocio permite a una empresa poder operar todo el tiempo. Más que un plan para la recuperación ante desastres, la continuidad del negocio es la estrategia general para garantizar que las operaciones continuarán en caso de desastres provocados por un error humano o de cualquier otra índole.

En la actualidad, los documentos electrónicos son el alma de la mayoría de las empresas y una herramienta ECM desempeña un papel importante en la continuidad del negocio. Las tecnologías ECM proporcionan repositorios centralizados donde la información corporativa vital puede guardarse. El método de almacenamiento puede variar en función de la importancia del contenido. Una planificación rigurosa de la continuidad de un negocio demuestra que no todo el contenido es fundamental. Los pasos a tener en cuenta en la planificación son los siguientes:

1. Dar prioridad a los contenidos para determinar con qué rapidez necesitan estar de nuevo en línea.
2. Determinar los procesos críticos y las entidades que los estén ejecutando.
3. Realizar una evaluación de impacto sobre las empresas para determinar el daño de una interrupción o pérdida en esos procesos.
4. Definir lo que una empresa considera un desastre y explicar cómo los procesos claves se recuperarán.
5. Establecer un centro de operaciones de crisis con los procedimientos para la cadena de mando y otras funciones.
6. Actualizar y poner a prueba el plan de recuperación anualmente o según las necesidades cambiantes del negocio.

Un plan de continuidad mejorará la capacidad de una organización para recuperarse durante una falla del sistema, además permitirá definir mejor la prioridad del contenido empresarial [10].

Teniendo en cuenta lo anterior, es posible asegurar que una de las estrategias para garantizar la continuidad son las copias de seguridad y los sistemas de gestión de contenido empresarial deben garantizar mecanismos para ello. De dicha manera se evidencia la necesidad de analizar cómo dichos sistemas realizan las copias de seguridad, ello permitiría elaborar una propuesta de solución que posibilite la implementación de algoritmos para visualizar, seleccionar y exportar los contenidos del

repositorio del ECM Alfresco Community 3.0 y así disminuir el espacio y tiempo de ejecución de sus copias de seguridad.

Antes es preciso definir en qué consisten los sistemas de gestión de contenido empresarial.

1.2. Sistemas de gestión de contenido empresarial

Los sistemas de gestión de contenido empresarial combinan diferentes tecnologías y componentes que también son útiles como soluciones autónomas, sin llegar a formar parte de un sistema empresarial integrado. Dichos componentes son los de captura, gestión, entrega, almacenamiento y preservación.

Captura: Contiene funcionalidades y tecnologías para la generación, captura, preparación y procesamiento de información analógica y electrónica. La captura puede incluir varios niveles de tecnología, desde la captura simple de información hasta la preparación compleja de información con la utilización de clasificación automatizada. Los componentes incluidos en la “captura” son frecuentemente llamados componentes de entrada [10].

Gestión: Los componentes de gestión sirven para la administración, procesamiento y utilización de la información. Para ello se utilizan funcionalidades que incluyen [10]:

- Bases de datos para la administración de la información.
- Sistemas de autorización de acceso para la protección de la información.

Entrega: Los componentes de entrega son los utilizados para presentar la información desde los componentes de gestión, almacenamiento y preservación. Dichos componentes también contienen funciones utilizadas para introducir información en otros sistemas tales como las de generación de archivos de salida con algún formato en específico o para lectura, por ejemplo: convertir o comprimir información para los componentes de almacenamiento y preservación.

Los componentes de entrega comprenden tres grupos de funciones y medios: tecnologías para la transformación, seguridad y distribución. Los servicios de transformación y seguridad deben estar disponibles para todos los componentes del sistema ECM de la misma forma.

Las funcionalidades en la categoría de entrega son también conocidas como salida y resumidas bajo el término de gestión de salida [10].

Almacenamiento: Los componentes de almacenamiento son utilizados para el almacenamiento temporal de la información, que no resulta necesaria que se archive. Igualmente, los medios empleados para este fin se pueden utilizar para un almacenamiento a largo plazo.

Los componentes de almacenamiento pueden ser divididos en tres categorías: repositorios como los lugares de almacenamiento, servicios de biblioteca como los componentes administrativos para los lugares de almacenamiento y tecnologías utilizadas para el almacenamiento [10].

Preservación: Los componentes de preservación se utilizan para el almacenamiento a largo plazo, estático e inamovible de la información, así como para el almacenamiento temporal de la información que no se desea o requiere archivar [10].

El objetivo de un sistema ECM integrado es ofrecer las funcionalidades antes mencionadas de manera unificada y como servicios para todos los subcomponentes de la gestión: gestión de documentos, colaboración, gestión de contenidos web, gestión de archivos y flujos de trabajo/gestión de procesos de negocio.

Para permitir la integración entre los diversos componentes que forman parte de la gestión, todos deben disponer de interfaces estandarizadas y procesos de transacción para la comunicación entre ellos.

Luego de definir qué es un sistema ECM y sus componentes se analizará cómo los mismos realizan las copias de seguridad. Para ello, como se verá en próximos epígrafes, se requiere la ejecución de copias de seguridad a sus sistemas de archivos y SGBD.

1.3. Análisis de las copias de seguridad en los ECM y los algoritmos utilizados para su ejecución

Para hacer un análisis de los sistemas de gestión de contenido empresarial, a efectos de la presente investigación, resulta interesante valorar de cada uno de ellos los siguientes aspectos:

- Forma en que realizan las copias de seguridad.
- Si emplean algún algoritmo para visualizar, seleccionar y exportar los contenidos a incluir en las copias de seguridad desde sus clientes web.

1.3.1. Copias de seguridad en los sistemas de gestión de contenido empresarial

Los sistemas de gestión de contenido empresarial pertenecen al conjunto de SGDEA, por lo que deben cumplir con los requisitos que aparecen en MoReq para las copias de seguridad. Además deben garantizar la continuidad, como define la gestión de contenido empresarial.

En el presente epígrafe se expone cómo los sistemas ECM, específicamente los open source, realizan las copias de seguridad; así como las principales ventajas y desventajas de los mismos a tener en cuenta para elaborar una propuesta de solución.

OpenKm

OpenKm es un ECM implementado en Java en su versión J2EE y se ejecuta sobre el servidor de aplicaciones JBoss.

Dicho sistema no incorpora ninguna funcionalidad, en su cliente web, para la realización de copias de seguridad. Las copias de seguridad en OpenKm dependen de la configuración existente en su repositorio, por defecto todos los documentos son guardados en la carpeta “\$JBASS_HOME/repository”, que puede ser modificada cambiando la propiedad “repository.home” en el archivo de configuración “OpenKM.cfg”.

Para realizar una copia de seguridad al repositorio de OpenKm es necesario hacer una copia de la carpeta antes mencionada. A continuación se muestran los pasos para la realización de una copia de seguridad [11]:

- Detener JBoss.
- Realizar una copia de seguridad del sistema de archivos, cuya ubicación la indica la propiedad “repository.home” en el archivo “OpenKM.cfg”.
- Realizar un backup de la base de datos.
- Iniciar JBoss.

OpenKm no ofrece una variante para ejecutar las copias de seguridad desde su interfaz visual, por lo que no propone una vía para incluir de manera selectiva los contenidos en las mismas. Necesariamente hay que exportar todo el repositorio desde la consola del SO.

Nuxeo

Nuxeo es un ECM programado en Java y puede ejecutarse sobre los servidores de aplicaciones JBoss y Tomcat, además es multiplataforma por lo que puede ser instalado en los SO Windows, Mac o Linux.

Este sistema de gestión documental tampoco incorpora, en su cliente web, ninguna funcionalidad para la realización de copias de seguridad y propone una serie de pasos, los que pueden ejecutarse en frío (sistema detenido) o en caliente (sistema ejecutándose). Para la realización de copias de seguridad es necesario ejecutar los siguientes pasos en la consola del SO de acuerdo al tipo de backup (frío o caliente) que se desee realizar:

Backup en frío:

1. Detener el servidor de aplicaciones.
2. Realizar un backup del sistema de archivos.
3. Realizar un backup de la base de datos.
4. Iniciar el servidor de aplicaciones.

Backup en caliente:

1. Realizar un backup de la base de datos.
2. Realizar un backup del sistema de archivos.

Si no se configura Nuxeo para utilizar una base de datos entonces se utilizará la base de datos Derby por defecto. Esta base de datos utiliza el sistema de archivos de Nuxeo por lo que en caso de realizar un backup se necesita detener el servidor de aplicaciones, es decir, todos los backup se realizarán en frío [12].

Nuxeo no ofrece una variante para ejecutar las copias de seguridad desde su interfaz visual, por lo que no propone una vía para incluir, de manera selectiva, los contenidos

en las mismas. Necesariamente hay que exportar todo el repositorio desde la consola del SO.

Knowledgetree

Se trata de un ECM con el que se puede estructurar la documentación en las empresas. Está programado en PHP y utiliza al SGBD MySQL; además necesita Apache para su ejecución.

Knowledgetree no posee ninguna funcionalidad en su cliente web que sea capaz de realizar copias de seguridad, no obstante a la hora de realizar dicha acción, desde la consola del SO, existen cuatro componentes que son necesarios incluir [13]:

- Base de datos: se le realiza el backup con el comando que MySQL utiliza para esta acción.
- Documentos: se encuentran ubicados, por defecto, en la raíz de la instalación de Knowledgetree en “var/Documents”.
- Archivos de configuración: el archivo de configuración es “config.ini” y se encuentra ubicado en la raíz de la instalación de knowledgetree dentro de “config/config.ini”.
- Aplicación: a pesar de que se puede realizar a partir de la instalación original, la realización de una copia de la aplicación permite una restauración más rápida y hace posible la restauración en caso de pérdida de la instalación original.

Con la utilización de Knowledgetree necesariamente hay que exportar todo el repositorio desde la consola del SO, no incorpora ninguna funcionalidad para incluir de manera selectiva los contenidos desde su cliente web. Ello provoca que independientemente de los contenidos que se desee incluir en la copia de seguridad siempre se exporte todo el repositorio.

Magnolia

Magnolia es un ECM especializado en manejo de contenido y documentos. Este sistema se encuentra implementado en Java, además utiliza el servidor de aplicaciones Tomcat. Posee dos ediciones, la Enterprise (propietaria) y la Community (totalmente libre).

Para la realización de copias de seguridad, Magnolia incluye en su edición Enterprise, a partir de la versión 3.6, un módulo capaz de realizar copias desde su cliente web. La copia de seguridad se le realiza al repositorio completo, la estructura del mismo en la carpeta donde se almacene es la siguiente [14]:

- repoConfig.zip: contiene los archivos de configuración necesarios para la restauración del repositorio de datos.
- history.gz: contiene una versión de los contenidos ubicados en los repositorios que contenga Magnolia.
- <workspace_name>.gz: contiene un archivo de este tipo por cada carpeta existente en el repositorio. Este archivo guarda todo el contenido del espacio y tiene el mismo nombre con que se encuentra en el repositorio.
- blobs: carpeta que contiene los binarios o contenidos de todos los espacios o carpetas.

Magnolia posibilita desde su cliente web realizar las copias de seguridad, no obstante no permite escoger el contenido a incluir en las mismas, necesariamente hay que exportar todo el repositorio.

Alfresco

Para realizar el análisis de las variantes en que el ECM Alfresco realiza las copias de seguridad sólo se tendrán en cuenta aquellas que se ejecutan desde el cliente web Alfresco Explorer, que es la interfaz utilizada en el marco de proyecto de Gestión Documental y Archivística al cual tributa el presente trabajo. Además se analizarán las variantes que aparecen en la página oficial de Alfresco.

A continuación se analizan las variantes encontradas:

1. En la página oficial de Alfresco existen una serie de pasos y acciones a realizar para la ejecución de la copia de seguridad de su repositorio. La variante propuesta tiene la limitante que se debe ejecutar desde la consola del SO haciendo uso de herramientas y comandos, tanto para realizar la copia de seguridad de la BD, como para el sistema de archivos y los índices de Lucene. La utilización de dichas herramientas y comandos no está al alcance de usuarios con conocimientos básicos de computación.

Independientemente de los documentos a los que se quiera realizar una copia de seguridad, siempre hay que incluir todo el repositorio de contenidos lo cual incrementa el espacio que ocupan las mismas en el dispositivo de almacenamiento. Además, el tiempo de ejecución no es proporcional al volumen total de documentos incluidos en la copia.

2. En el año 2008 Intecna Soluciones [15] desarrolló un módulo en Alfresco para la Junta de Andalucía [16] que entre otras, presenta las siguientes características:

- Recuperación parcial y selectiva de los contenidos: a la hora de realizar la restaura de la copia de seguridad el usuario puede escoger cuáles documentos o contenidos desea incluir en ella.
- Definición y configuración de tareas de exportación automática: el usuario puede programar la fecha, hora y frecuencia con que desea realizar una copia de seguridad a las carpetas que contengan el o los contenidos que desee incluir en la misma.
- Asistente para realizar la configuración de tareas, restauración y ejecución de los backup: para su realización dichas tareas presentan un grupo de páginas (wizard) que guían al usuario en su ejecución.
- Backup realizado en almacenamiento externo: las copias de seguridad las almacena fuera del repositorio de Alfresco, es decir, en algún dispositivo de almacenamiento (disco duro, memoria flash, cinta magnética, etc.).
- La ejecución del backup es una acción asociada a las carpetas: al ejecutar una copia de seguridad no se puede almacenar el contenido directamente, hay que almacenar la carpeta que lo contiene, aunque la misma contenga otros contenidos los cuales el usuario no desee incluir en dicha copia.

3. Desde el cliente web Alfresco Explorer, el ECM Alfresco Community 3.0, da la posibilidad de exportar e importar su contenido utilizando los Paquetes de Contenidos de Alfresco (Alfresco Content Package, en inglés, abreviado ACP). Los ACP [4] son archivos ZIP⁴ que cuentan con la extensión (.acp) e incluyen en él a los metadatos y contenidos a exportar.

⁴ Archivos simples que contienen uno o varios archivos comprimidos.

A continuación se realiza un análisis crítico sobre cada una de las variantes enumeradas:

El módulo desarrollado por Intecna, al integrarse al cliente web Alfresco Explorer, desde el punto de vista del aprovechamiento de espacio en el dispositivo de almacenamiento, ofrece mejores prestaciones con respecto a la variante que se ofrece en la página oficial de Alfresco, pues la ejecución del backup está asociada a las carpetas. En la figura 1.1, se muestra la interfaz brindada por el módulo de Intecna para seleccionar el espacio a exportar:



Figura 1.1 Interfaz visual que permite seleccionar el espacio a exportar del módulo desarrollado por Intecna [17].

La manera en que se selecciona el espacio, ilustrado en la figura 1.1, brindada por el módulo de Intecna provoca que disminuya el espacio a utilizar por la copia de seguridad en el dispositivo de almacenamiento, en proporción a los archivos incluidos en la misma. Si se desea realizar una copia a algún archivo en específico se debe incluir en el backup al espacio o carpeta que lo contiene y no así a todo el repositorio, como se propone que se realice en la página oficial del ECM Alfresco.

No obstante dicho módulo presenta varios inconvenientes:

1. Si bien disminuye el espacio a utilizar por la copia de seguridad en el dispositivo de almacenamiento, aún este indicador no es el apropiado acorde al tamaño de los documentos que se quieren exportar; dicho módulo no permite escoger

específicamente el contenido a incluir en el backup. Necesariamente se debe agregar el espacio que los contiene, lo que implica la realización de una copia a todos los contenidos que se encuentran en él.

De esta manera aún se incluyen en la copia de seguridad documentos o contenidos no deseados, lo que provoca la utilización extra de espacio en el dispositivo de almacenamiento así como en su tiempo de ejecución.

2. No permite incluir en las copias de seguridad los archivos de configuración del ECM Alfresco.
3. El módulo de Intecna está desarrollado para el cliente web Alfresco Explorer del ECM Alfresco 2.0 y 2.1, por lo que no es compatible con el cliente web Alfresco Explorer del Alfresco Community 3.0.

Como principal ventaja tiene:

- Al importar los contenidos propone una vía para incluirlos de manera selectiva.

La variante de exportar e importar contenidos utilizando los ACP desde el cliente web Alfresco Explorer presenta las siguientes ventajas y desventajas:

Ventajas:

- La utilización de los ACP elimina la necesidad de realizar copias de seguridad a la BD y al sistema de archivos del ECM Alfresco desde la consola del SO. Los ACP incluyen los metadatos y contenidos a exportar.

Desventajas:

- La acción se encuentra asociada al espacio, no permite incluir un documento en específico. Necesariamente hay que exportar al espacio que lo contiene. Lo anterior se ilustra en la figura 1.2.
- No permite programar la creación automática de ACP.
- No permite exportar los ACP fuera del repositorio de Alfresco, como se ilustra en la figura 1.2.

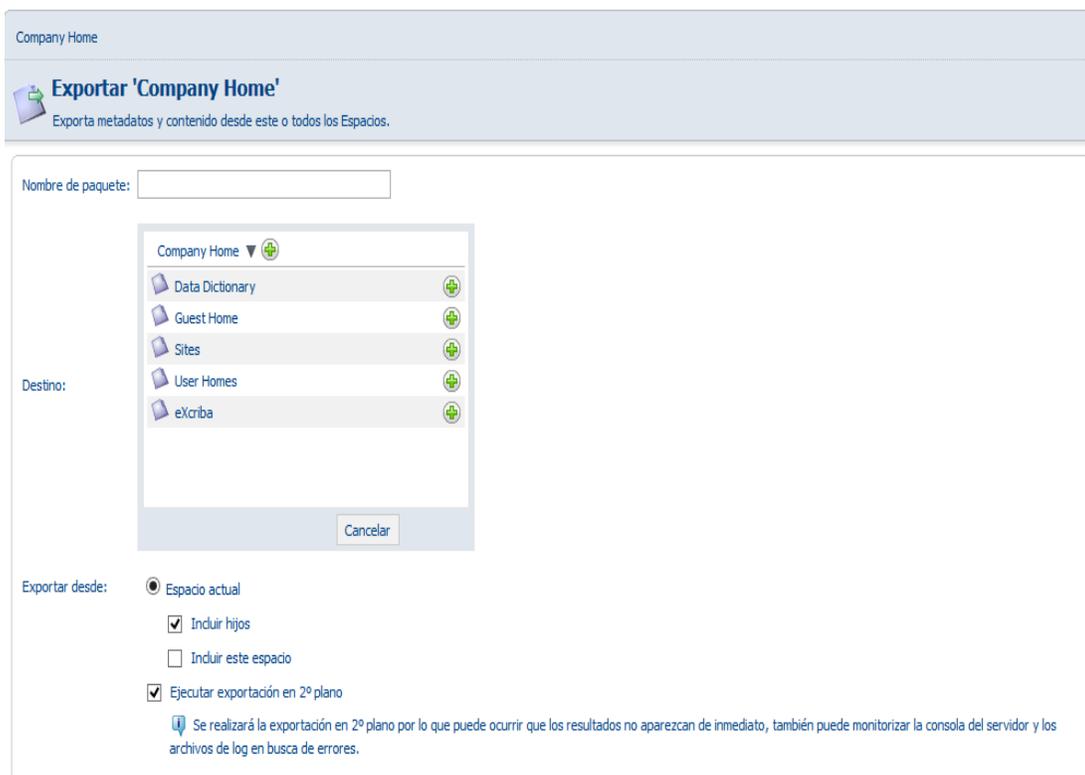


Figura 1.2 Interfaz visual que permite exportar un espacio desde el Alfresco Explorer.

En la figura 1.2 se muestra la interfaz brindada por el Alfresco Explorer del Alfresco Community 3.0 para exportar el espacio “Company Home”. Se observa que en “Destino” se puede seleccionar los espacios donde almacenar el ACP generado por “Company Home”.

1.3.2. Algoritmos para visualizar, seleccionar y exportar, desde un cliente web, los contenidos del repositorio

Con el objetivo de realizar una propuesta de algoritmos para la visualización, selección y exportación, desde el cliente web Alfresco Explorer, de los contenidos ubicados en el repositorio del ECM Alfresco Community 3.0 se hace necesario realizar un estudio de cómo los algoritmos utilizados por los ECM analizados en el epígrafe anterior realizan dichas funcionalidades desde sus clientes web.

OpenKm, Nuxeo y Knowledgetree: no implementan algoritmos de visualización, selección y exportación del contenido a incluir en las copias de seguridad desde sus clientes web. Realizan las copias a sus sistema de archivos y base de datos con las herramientas y comandos que brindan los SO para dichas funcionalidades.

Magnolia: aunque posibilita desde su cliente web realizar copias de seguridad, no implementa algoritmos para la visualización y selección del contenido a incluir en las mismas, exporta todo el repositorio.

Respecto a la exportación, no se dispone del código fuente para así analizar la forma en que realiza dicha funcionalidad, la misma está incluida en la edición Enterprise (propietaria).

Alfresco:

1. El Alfresco Explorer implementa un mecanismo de copias de seguridad mediante la creación de ACP. Dicho mecanismo permite crear, dentro del propio repositorio del ECM Alfresco, un comprimido de extensión (.acp) con el contenido. Esta variante no propone ningún algoritmo para visualizar y seleccionar el contenido a incluir en la copia de seguridad pues es una acción asociada a un espacio. Siempre se exporta todo el contenido ubicado en el espacio seleccionado.

En cuanto al algoritmo para exportar el contenido, el mecanismo de creación de los archivos (.acp) utilizado por Alfresco se puede reutilizar en la propuesta de solución. En este caso, habría que redefinirla para poder crear dichos archivos fuera del repositorio del ECM Alfresco Community 3.0.

2. El módulo desarrollado por Intecna, pese a no poder integrarse al Alfresco Community 3.0, ofrece una variante a utilizar para la visualización y selección del contenido. El módulo desarrollado visualiza los contenidos a importar mediante una estructura de tipo árbol [18].

Las funcionalidades presentes en el módulo desarrollado por Intecna utilizan los servicios implementados para el Alfresco 2.0 los cuales, en algunos casos, no son los mismos que se implementaron para el Alfresco 3.0 provocando, en ocasiones, el poco entendimiento del código fuente. Además, el árbol generado por el módulo de Intecna para la importación del contenido se genera a partir de ficheros que contienen información básica o índices asociados a los espacios exportados [17].

En el caso de la propuesta de solución a desarrollar es necesario que el árbol generado, para así visualizar el repositorio, sea a partir de los contenidos existentes en

el mismo, independientemente que hayan sido exportados o no. Para lograr lo antes planteado se necesita recorrer estructuras de tipo árbol por lo que a continuación se analizarán los algoritmos existentes para ello.

1.4. Algoritmos para recorrer estructuras de tipo árbol

En el presente epígrafe se exponen las diferentes variantes de algoritmos para recorrer estructuras de tipo árbol de forma tal que sea posible seleccionar cuál es la más adecuada para la propuesta de solución de la presente investigación.

Para recorrer las estructuras de tipo árbol se utilizan métodos recursivos [18] [19], dada su propia definición. Existen dos tipos de recorridos, en profundidad (preorden, entreorden y postorden) y a lo ancho [20]. En los recorridos en profundidad cada vez que se visita un nodo el próximo a visitar tiene una profundidad mayor que el actual y en los recorridos a lo ancho se visitan los nodos en el orden de los niveles del árbol.

A continuación se muestra el pseudocódigo de los algoritmos para cada uno de los recorridos:

```
PreOrden(L)  
  Si grado = 0 entonces  
    L.Adicionar(raiz);  
  Sino  
    L.Adicionar(raiz)  
  Desde i = 1 Hasta grado Con Paso 1 Hacer  
    Subárbol[i].PreOrden(L)  
  Fin // Sino  
fin // Preorden
```

Figura 1.3 Algoritmo recursivo de recorrido en preorden [20].

```

EntreOrden(L)
    Si grado = 0 entonces
        L.Adicionar(raiz);
    Sino
        Subárbol[0].EntreOrden(L)
        L.Adicionar(raiz)
        Desde i = 1 Hasta grado Con Paso 1 Hacer
            Subárbol[i].EntreOrden(L)
    Fin // Sino
fin // EntreOrden

```

Figura 1.4 Algoritmo recursivo de recorrido en entreorden.

```

PostOrden(L)
    Si grado = 0 entonces
        L.Adicionar(raiz);
    Sino
        Desde i = 1 Hasta grado Con Paso 1 Hacer
            Subárbol[i].PostOrden(L)
        L.Adicionar(raiz);
    Fin // Sino
fin // PostOrden

```

Figura 1.5 Algoritmo recursivo de recorrido en postorden.

```

A_Lo_Ancho(Lista L)
    Crear ColaAux;
    ColaAux.Adicionar(arbol);
    Mientras ! ColaAux.Vacia()
        arbolActual = ColaAux.Extraer()
        L.Adicionar(arbolActual.Raiz())
        Desde i=1 Hasta arbolActual.Grado() Con Paso 1 Hacer
            ColaAux.Adicionar(arbolActual.Subarbol[i])
    FIN //Mientras
FIN // A_Lo_Ancho

```

Figura 1.6 Algoritmo de recorrido a lo ancho [20].

Si los algoritmos anteriores se utilizaran para visualizar y exportar el repositorio de Alfresco recorrerían cada uno de los contenidos del repositorio. Ello constituye un inconveniente si sólo se desea visualizar o exportar un determinado grupo de

contenidos y no todo el repositorio pues conlleva a demoras en el tiempo de procesamiento del algoritmo. Por lo antes planteado surge la necesidad de proponer un algoritmo que permita visualizar únicamente los contenidos seleccionados por el usuario y otro que sólo recorra las carpetas que contengan contenidos seleccionados por el usuario para su exportación.

1.5. Herramientas y tecnologías

Debido a la necesidad de implementar los algoritmos que se utilizarán en la propuesta de solución para visualizar y exportar el contenido es necesario realizar un análisis de las herramientas, tecnologías y metodologías que se emplearán en dicha implementación, así como para integrarlos al cliente web Alfresco Explorer.

Metodología de desarrollo:

- El proceso Unificado de Software (Rational Unified Process, en inglés, abreviado RUP) es una metodología de desarrollo de software robusta, la cual, entre todas las metodologías estándares, es una de la más utilizadas para el diseño, implementación y documentación de sistemas orientados a objetos, esta no presenta pasos que estén firmemente establecidos sino que cuenta con un conjunto de flujos de trabajo adaptables al contexto y necesidades de cada organización. RUP cuenta con un ciclo de vida que se caracteriza por ser [21]: dirigido por casos de uso, centrado en la arquitectura e iterativo–incremental. Además de las características antes mencionadas la utilización de esta metodología de desarrollo está avalada por ser la empleada en el marco del proyecto de Gestión Documental y Archivística.

Lenguaje unificado de modelado:

- (Unified Model Language, en inglés, abreviado UML) es un lenguaje utilizado para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de *software* [22], es de los más utilizados ya que ofrece un estándar para describir un plano o modelo del sistema en el cual se incluyen los conceptos de proceso de negocio así como funciones del sistema. UML en su funcionamiento emplea los siguientes tipos de diagramas [22]: de estructura estática (clases, objetos y casos de uso), de comportamiento (interacción, estado, actividad) y de implementación (componentes, despliegue).

Herramienta CASE (por sus siglas en ingles: Computer Aided Software Engineering)

- Visual Paradigm es una suite de trabajo (entiéndase por suite: varias herramientas dentro de una sola) fácil de utilizar, la cual cuenta con herramientas para el diseño de diagramas y los artefactos que se generan durante los ciclos de desarrollo de software. Permite la integración con varios Entornos de Desarrollo Integrado (Integrated Development Enviroment, en inglés, abreviado IDE) como por ejemplo: NetBeans y Eclipse. En la UCI se cuenta con la licencia para esta suite, lo cual facilita las condiciones de uso [23].

Lenguajes de Programación:

- Java: es el utilizado por Alfresco para su implementación, es un lenguaje de programación orientado a objetos, libre y portable. La portabilidad viene dada por el hecho de que el compilador del lenguaje genera un código binario (bytecode) el cual es interpretado por la Máquina Virtual de Java (Java Virtual Machine, en inglés, abreviado JVM), por lo que un programa escrito en Windows puede ser interpretado en un entorno libre, con solo disponer de la máquina virtual para dichos entornos [24].
- JavaScript: es un lenguaje de script multiplataforma orientado a objetos, es pequeño y ligero; está diseñado para ser utilizado en otros productos y aplicaciones tales como los navegadores web. Es utilizado por Alfresco para reconocer y tratar, en el cliente (navegador web), los eventos generados por el usuario [25].

Frameworks:

- Spring: framework de código abierto para el desarrollo de aplicaciones en Java y es utilizado para la implementación de todos los componentes y servicios existentes en Alfresco.
- JavaServer Faces (JSF): framework para el desarrollo de aplicaciones web en Java y es el utilizado para la implementación de las interfaces web de Alfresco.

IDE:

- Eclipse: es el IDE con el cual se desarrolló el Alfresco Community 3.0 por lo que su código fuente, por defecto, se puede importar en él; debido a ello, para

implementar el módulo propuesto se utilizará Eclipse, específicamente el Eclipse Helios.

Conclusiones parciales

1. Los ECM analizados no incluyen desde sus clientes web funcionalidades para visualizar, seleccionar y exportar los contenidos de sus repositorios. Para llevar a cabo la exportación es necesario realizar la copia de seguridad al sistema de archivos y base de datos en su totalidad.
2. Seleccionando los contenidos a incluir en las copias de seguridad del ECM Alfresco Community 3.0, desde el cliente web Alfresco Explorer, se puede disminuir el espacio a ocupar por las mismas en el dispositivo de almacenamiento así como su tiempo de ejecución para una determinada cantidad de contenidos.
3. Para exportar desde el cliente web Alfresco Explorer del Alfresco Community 3.0 los contenidos seleccionados es necesario implementar y adicionar nuevas funcionalidades con algoritmos que permitan visualizar, seleccionar y exportar los contenidos del repositorio a partir de una estructura de tipo árbol.
4. Las metodologías y tecnologías más adecuadas para diseñar e implementar los algoritmos que permitan visualizar, seleccionar y exportar los contenidos del repositorio, así como para integrarlos al cliente web Alfresco Explorer son: Java y JavaScript como lenguaje de programación, Spring y JSF como marco de trabajo y RUP como metodología para el desarrollo.

2. Propuesta de solución para la Gestión de Copias de Seguridad en el ECM Alfresco Community 3.0

Con el objetivo de lograr que el tiempo de ejecución y el espacio que ocupan las copias de seguridad sean directamente proporcional al tamaño total de los documentos incluidos en la misma, además de mejorar la usabilidad con que se realizan en el ECM Alfresco Community 3.0, se propone incorporar a su cliente web Alfresco Explorer un conjunto de algoritmos que posibiliten visualizar y exportar su repositorio. Permitiendo, de esta manera, incluir selectivamente los contenidos a exportar en la copia de seguridad. Para ello, en el presente capítulo, se proponen los algoritmos utilizados para representar los contenidos del repositorio en forma de una estructura jerárquica, así como los creados para exportar los contenidos fuera del repositorio de Alfresco. Además se muestra la arquitectura de la propuesta de solución.

2.1. Algoritmo propuesto para la visualización del contenido

Para lograr incluir de manera selectiva los contenidos a exportar en la copia de seguridad es primordial visualizar el repositorio de Alfresco. Para ello es necesario utilizar algún algoritmo que permita mostrar los contenidos que luego se desean exportar.

En el epígrafe 1.4 se analizaron los algoritmos para recorrer las estructuras de tipo árbol. De los algoritmos analizados se puede concluir que el apropiado para visualizar el componente visual con la representación del repositorio de Alfresco es el que realiza el recorrido en preorden debido a que, inicialmente, cada uno de los nodos del árbol tiene que ser un espacio del repositorio que a su vez puede contener varios espacios, hasta encontrar espacios vacíos o que incluyan solamente documentos. Para ayudar a entender lo explicado anteriormente se muestra a continuación la figura 2.1, que lo ilustra visualmente:

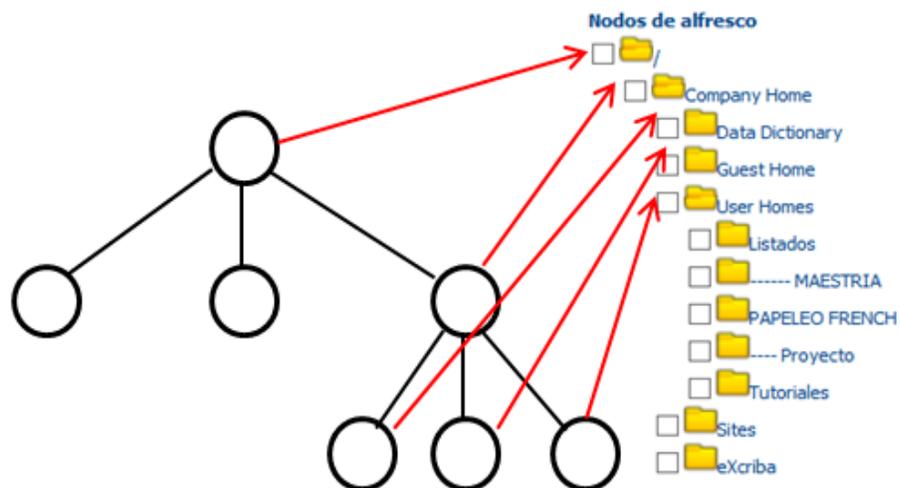


Figura 2.1 Relación entre los nodos del árbol y los espacios del repositorio de Alfresco.

Una variación del algoritmo que realiza el recorrido en preorden para así recorrer el repositorio de Alfresco y representarlo en el componente visual es el que se muestra a continuación en la figura 2.2:

```

construirRecursivo(Nodo)// inicialmente la variable Nodo toma el valor del nodo raíz de Alfresco
agregarAlArbol(Nodo) // se agrega una carpeta al árbol que visualiza el repositorio
  hijos = listaDeHijos(Nodo) //devolver la lista de hijos de Nodo
  Para i = 1, ...,cantidadHijos //ciclo para cada hijo de Nodo1
    Si hijos[i] es carpeta
      construirRecursivo(hijos[i]) //llamada recursiva
    Fin Si
    Si hijos[i] no es carpeta
      agregarAlArbol(hijos[i]) //se agrega un documento al árbol que visualiza el repositorio
    Fin si
  Fin para(i)
Fin //construirRecursivo

```

Figura 2.2 Pseudocódigo del algoritmo para recorrer y mostrar el repositorio de Alfresco en preorden.

Si se representara el componente visual con los contenidos de todo el repositorio de Alfresco utilizando el algoritmo mostrado en la figura 2.2, para una elevada cantidad de contenidos almacenados en el repositorio, puede resultar muy costoso desde el punto de vista del tiempo de ejecución.

Por dicho motivo se propone un nuevo algoritmo, mostrado en la figura 2.3, que será utilizado para presentar el componente visual, que sólo muestra los contenidos hijos de

la carpeta o espacio que el usuario decida expandir o visualizar en el componente visual.

```
mostrarHijosDe(nodoArbolVisual, nodoAlfresco) // nodos del componente visual y el repositorio de Alfresco  
    carpetasHijas = listaDeCarpetas(nodoAlfresco) //devuelve listado de carpetas hijas  
    Para i = 1, ..., cantidadCarpetas  
        agregarAlArbolHijosDe(nodoArbolVisual, carpetasHijas[i]) // 1  
    Fin para(i)  
    archivosHijos = listaDeArchivos(nodoAlfresco) //devuelve listado de archivos hijos  
    Para j = 1, ..., cantidadArchivos  
        agregarAlArbolHijosDe(nodoArbolVisual, archivosHijos[j]) // 2  
    Fin para(j)  
Fin //mostrarHijosDe
```

Figura 2.3 Pseudocódigo del algoritmo para mostrar los hijos de un nodo seleccionado en el componente visual.

En el algoritmo mostrado en la figura 2.3, en (1) se agregan al componente visual todos los nodos carpetas que son hijos de la carpeta que se quiere visualizar. En (2) se agregan al componente visual los nodos archivos que son hijos de la carpeta que se escoge visualizar. Es de destacar que el algoritmo utilizado no es recursivo debido a que cuando se escoge visualizar una carpeta, sólo se muestran los espacios o documentos que son descendientes (hijos) directos de la carpeta seleccionada en el componente visual.

En la figura 2.4 se muestra la carpeta “Company Home” sobre la cual, luego de ser seleccionada para su visualización, se aplicará el algoritmo antes mencionado:



Figura 2.4 Componente que visualiza el repositorio en forma de árbol con el nodo raíz y Company Home como hijo del mismo.

A partir de lo mostrado en la figura 2.4 si se selecciona la carpeta “Company Home” se muestran los hijos de la misma, los que son mostrados en la figura 2.5:

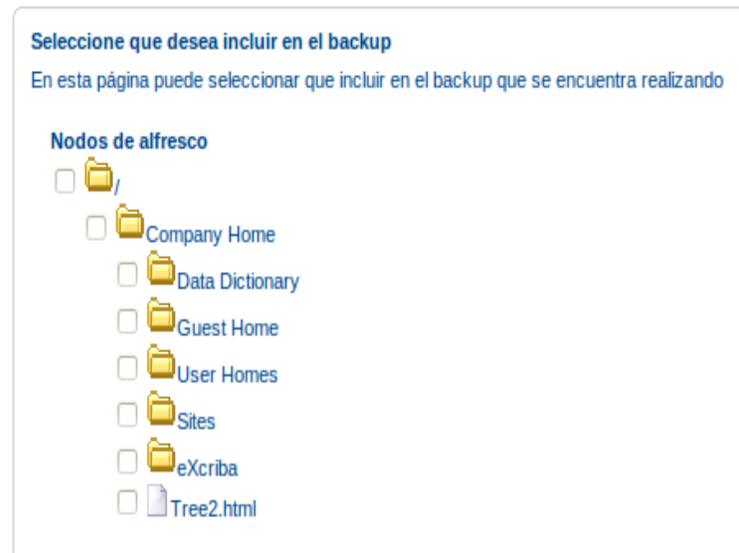


Figura 2.5 Componente que visualiza el repositorio de Alfresco con los hijos de Company Home.

El algoritmo implementado permite representar en un componente visual, mediante una estructura de tipo árbol, los hijos de los espacios seleccionados, lo que posibilita disminuir el tiempo de respuesta de dicho componente que visualiza el repositorio de Alfresco.

2.2. Algoritmos propuestos para exportar el contenido

En el epígrafe 1.4 se presentaron los algoritmos para recorrer estructuras de tipo árbol, si se utiliza alguno de ellos para recorrer un árbol que represente el repositorio de Alfresco, necesariamente, se visitarían cada uno de los nodos del árbol. Lo anterior es un inconveniente pues para una elevada cantidad de nodos se hace muy costoso el recorrido del árbol, en lo que a tiempo de ejecución del algoritmo respecta, por lo que sería conveniente disminuir la cantidad de nodos que se visitan cuando se recorre el árbol.

Si se limita la cantidad de carpetas que contengan carpetas o documentos a exportar que debe procesar el algoritmo, con el objetivo de incluir en la copia de seguridad sólo los documentos y espacios que se encuentren seleccionados, disminuye el tiempo de ejecución del algoritmo. Para ello es necesario realizar una variación a alguno de los algoritmos recursivos utilizados en el recorrido de estructuras de tipo árbol para que sólo se recorran los nodos (carpetas) si tienen algún nodo descendiente (carpeta o

archivo) seleccionado para la exportación. A continuación se muestra en la figura 2.6 el algoritmo creado:

```
exportarArbol( nodo ) //se inicializa con el nodo raíz del árbol en el componente visual
  hijos = listaDeHijosSeleccionados ( nodo ) //1
  Para i = 1, .., cantidadHijos
    Si hijos[i] es carpeta
      Si hijos[i].CantidadMarcados == 1 y hijos[i].Seleccionado //2
        exportar( hijos[i] ) //3
      Sino Si hijos[i].CantidadMarcados >= 1 // 4
        exportarArbol( hijos[i] ) //5
      Fin Si
    Fin Si
  Fin Si
  Si hijos[i] es archivo y está marcado
    exportar( hijos[i] ) //6
  Fin si
Fin para(i)
Fin //exportarArbol
```

Figura 2.6 Pseudocódigo del algoritmo para recorrer los nodos seleccionados en el componente visual.

Del pseudocódigo mostrado en la figura 2.6 es necesario comentar que en (1) se devuelve el listado de hijos seleccionados para la exportación del nodo. Si el espacio en el componente fue seleccionado para la exportación entonces tendrá el contador de hijos seleccionados en uno (2) por lo que es necesario recorrer el repositorio de Alfresco para exportar todos sus descendientes (3). Si el nodo tiene algún descendiente marcado (4), entonces tendrá el contador de hijos marcados con un valor mayor o igual a uno por lo que es necesario recorrer los hijos de dicho espacio en el componente visual (árbol) para lo cual se hace una llamada al método recursivo (5). Al final, si algún hijo del nodo en el componente visual se encuentra marcado y es un archivo significa que hay que exportarlo (6).

En la figura 2.7 se muestra el componente visual con una representación del repositorio de Alfresco que se utilizará como ejemplo para ilustrar el recorrido con el algoritmo creado:



Figura 2.7 Componente que visualiza el repositorio de Alfresco.

Si el algoritmo creado es aplicado al árbol ilustrado en la figura 2.7, sólo se visitará la carpeta “Company Home” y dentro de ella se exportará todo el contenido que descienda de la carpeta “Guest Home”.

El algoritmo mostrado en la figura 2.6 sólo recorre las carpetas antes mencionadas debido a que previamente se redefine la clase que modela al componente visual y se le agrega, a cada nodo, un contador para indicar la cantidad de hijos que tiene seleccionados y una lista donde cada uno de sus elementos contiene un identificador y la referencia del nodo.

Cuando el usuario selecciona una carpeta o documento para su exportación, se actualiza dicha lista en cada uno de los espacios predecesores con el identificador del espacio y la referencia del mismo, además se aumenta en uno el contador que indica la cantidad de hijos seleccionados. Si se desmarca la carpeta o documento se decrementa el contador en cada uno de los espacios predecesores, si el contador es cero significa que el espacio no contiene ningún hijo para su exportación por lo que se elimina de la lista el identificador y referencia del nodo.

Cuando el algoritmo recorre los espacios seleccionados para la exportación, si el contador que indica la cantidad de hijos visitados es uno, exporta el espacio en su totalidad, si es mayor que uno significa que es necesario recorrerla pues contiene espacios hijos seleccionados para la exportación.

Este algoritmo presenta la ventaja que cada espacio del árbol recorre única y exclusivamente los espacios hijos seleccionados para la exportación. La lista de espacios a exportar que contiene cada uno de los mismos, actualizada cada vez que se marca o desmarca un espacio lo garantiza.

A partir del algoritmo mostrado en la figura 2.6, en (5) se implementa un nuevo algoritmo que utiliza la programación en paralelo [26] para, a la hora de exportar los documentos fuera del repositorio de Alfresco, disminuir el tiempo de ejecución de la copia de seguridad. Cada carpeta a exportar, si contiene documentos, se ejecuta en un hilo [27] donde, por cada uno de los documentos, se creará un ACP y se exportará fuera del repositorio de Alfresco. De esta manera se elimina el procesamiento secuencial de los contenidos a exportar.

Con el procesamiento en paralelo se procura elevar la velocidad de exportación, no obstante, es importante controlar la cantidad de hilos que se ejecutan.

Cuando se define una cantidad constante de hilos a levantar por el módulo no se tiene en cuenta si se puede ejecutar una cantidad mayor de hilos para elevar el tiempo de ejecución de las copias sin que disminuya el rendimiento del ECM Alfresco. Además, tampoco se tiene en cuenta si no es conveniente ejecutar la cantidad de hilos seleccionada producto a que el tiempo de las copias aumenta como resultado de la disminución del rendimiento del ECM Alfresco.

Por lo antes planteado se hace necesario crear un algoritmo que regule la cantidad de hilos a levantar cada vez que se ejecute una copia de seguridad en función de su tiempo de ejecución, cantidad y tamaño total de los documentos a incluir en la misma. El algoritmo creado, cada cinco copias, aumenta en uno la cantidad de hilos a levantar hasta llegar a una cantidad máxima de hilos determinada por el número de procesadores que tenga la computadora donde se ejecuten las copias. Cada cinco copias se descartan la de mayor y menor tiempo debido a que pueden ser picos de máximo o mínimo procesamiento, con las tres restantes se calcula el tiempo promedio

de ejecución. Cada uno de los tiempos son comparados entre sí, guardándose el menor y la cantidad de hilos con que se logró el mismo. Con lo anterior se logra que, cuando se alcanza la cantidad máxima de hilos, las copias se ejecuten con el número de hilos con que se logró el menor tiempo para la cantidad y tamaño total de documentos pasados como parámetros al algoritmo.

En la figura 2.8 se muestra el pseudocódigo del algoritmo antes mencionado:

```

cantidadHilosALevantar(cantidadDoc, tamañoDoc)
    listadoCopias = devolverListadoCopias( cantidadDocumentos, tamañoDocumentos ) //1
    adicionarHilos = (listadoCopias.Longitud / 5) //2
    MAX_COPIAS_PRUEBAS = (CANT_MAX_HILOS-1) *5
    mejoresValores = mejoresValores( listadoCopias ) //3
    Si (listadoCopias.Longitud % 5 == 0) y (listadoCopias.Longitud <= MAX_COPIAS_PRUEBAS)
        ultimasCincoCopias = devolverCincoUltimasCopias( listadoCopias ) //4
        auxMejoresValores = descartarPicos(ultimasCincoCopias) //5
        Si auxMejoresValores.Tiempo < mejoresValores.Tiempo //6
            mejoresValores = auxMejoresValores
        Fin Si
    Fin Si
    Si listadoCopias.Longitud < MAX_COPIAS_PRUEBAS //7
        hilosProxCopia = CANT_INICIAL_HILOS + adicionarHilos //8
    Sino
        hilosProxCopia = mejoresValores.hilos_A_Levantar //9
    Fin Si
Devolver hilosProxCopia
Fin //cantidadHilosALevantar

```

Figura 2.8 Pseudocódigo del algoritmo para determinar la cantidad de hilos a levantar.

Del algoritmo mostrado en la figura 2.8 se tiene que inicialmente almacena la información asociada a las últimas copias de seguridad realizadas con una cantidad y tamaño de documentos semejantes a las pasadas como parámetros (2). En “adicionarHilos” se guarda el número de hilos a adicionar en función de la cantidad de copias de seguridad realizadas.

El método utilizado en (3), cada cinco copias de seguridad descarta la de menor y mayor tiempo, de las tres restantes calcula su tiempo promedio. Cada uno de los anteriores tiempos son comparados entre sí y se devuelve el menor junto al número de hilos con que se alcanzó.

Posteriormente si la cantidad de copias de seguridad realizadas es múltiplo de cinco, se toman las cinco últimas (4) y de ellas se descarta la de menor y mayor tiempo, de las tres restantes se calcula y devuelve su promedio junto a la cantidad de hilos con que se logró (5). Los tiempos devueltos en (5) y (3) son comparados y se guarda el menor ellos junto a la cantidad de hilos correspondiente.

Si se realiza una cantidad de copias superior a las necesarias para establecer la cantidad de hilos a levantar, en función de la cantidad de procesadores representado por la variable "MAX_COPIAS_PRUEBAS", el algoritmo devuelve un número de hilos igual al que le corresponde el menor tiempo de ejecución alcanzado para la cantidad y volumen de documentos pasados como parámetros (9). De lo contrario la cantidad de hilos a levantar por la próxima copia debe ser igual a la cantidad inicial, almacenada en "CANT_INICIAL_HILOS", sumada a una cantidad adicional en dependencia del número de copias realizadas, almacenado en la variable "adicionarHilos" (8).

El número de copias seleccionadas para aumentar la cantidad de hilos es cinco debido se considera un número de copias considerable para determinar el rendimiento del ECM Alfresco en función de una determinada cantidad de hilos.

El algoritmo creado para recorrer y exportar el repositorio disminuye la cantidad de carpetas recorridas en el árbol que representa el repositorio de Alfresco. La exportación de las carpetas seleccionadas se realiza mediante un algoritmo creado para levantar un hilo de ejecución por cada carpeta a exportar, lo que trajo aparejado la necesidad de crear un nuevo algoritmo que regule la cantidad de hilos a levantar en cada exportación en dependencia de la cantidad y tamaño de los contenidos incluidos en la copia de seguridad así como de su tiempo de ejecución.

2.3. Creación del AMP para integrar los algoritmos de visualización y exportación del repositorio del ECM Alfresco Community 3.0 desde el cliente web Alfresco Explorer

Los algoritmos, antes propuestos, para la visualización y exportación del contenido se podrán integrar al cliente web Alfresco Explorer del ECM Alfresco Community 3.0 a través de un archivo para Módulo de Paquetes de Alfresco (Alfresco Module Package, en inglés, abreviado AMP) [28].

2.3.1. Integración de los algoritmos para la visualización y exportación en el AMP

El AMP propuesto permitirá a los administradores de Alfresco salvar y restaurar los contenidos y archivos de configuración fuera de su repositorio. Para ello es necesario seguir una serie de pasos y configuraciones [28] en el IDE donde se desarrollará el AMP para que el mismo sea creado e integrado al Alfresco.

Para integrar los algoritmos así como el resto de las funcionalidades en el AMP a desarrollar se utilizará el patrón de arquitectura Modelo Vista Controlador (MVC). Este patrón separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes (Modelo, Vista y Controlador). Cada una de estas clases son las utilizadas por Alfresco para extender sus funcionalidades donde Spring se utiliza para la implementación de la lógica en Java (Clases Controladoras) y JavaServer Faces para la implementación de las vistas o interfaces. Esto unido a los servicios brindados por Alfresco para acceder a su repositorio de contenido (Modelo) posibilita la independencia entre estas tres entidades a la hora de su implementación.

En la figura 2.9 se muestra la arquitectura a utilizar para poder integrar los algoritmos implementados al cliente web Alfresco Explorer del Alfresco Community 3.0:

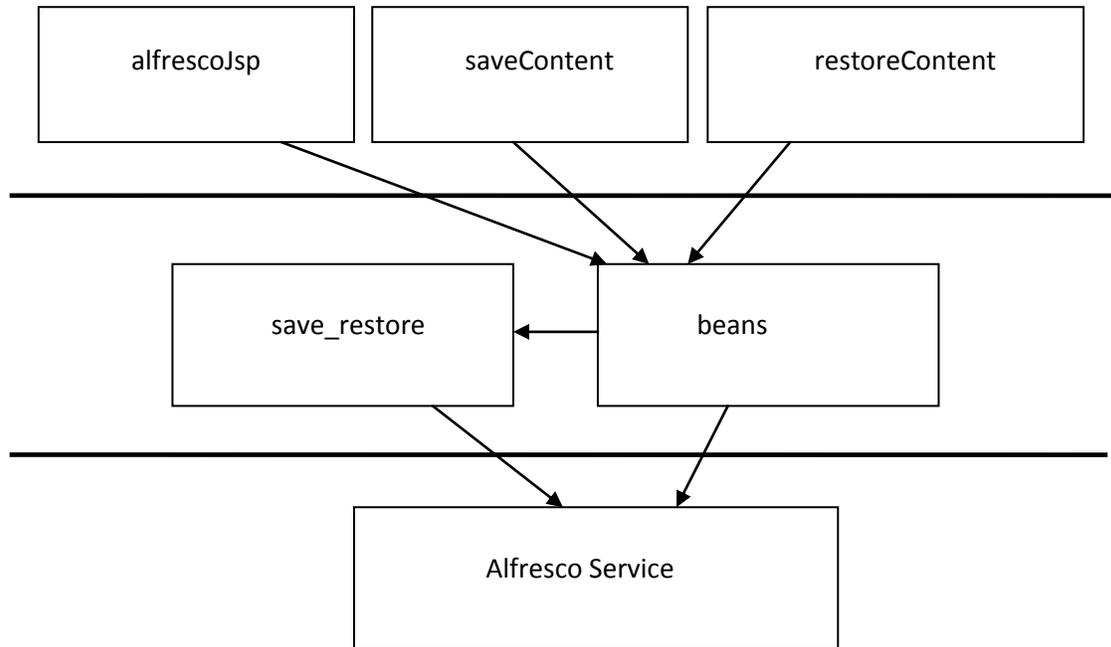


Figura 2.9 Arquitectura para el AMP de Gestión de Copias de Seguridad.

En la figura 2.9 se tiene que la vista está compuesta por los paquetes `alfrescoJsp`, `saveContent` y `restoreContent`, al controlador pertenecen los paquetes `save_restore` y `beans`. El modelo está conformado por los servicios que brinda Alfresco para acceder a su repositorio de contenido.

Los algoritmos implementados para la visualización y exportación del repositorio se incluyen al AMP de la siguiente manera:

- Los algoritmos para la visualización del repositorio (figura 2.3) y para recorrer el componente visual con la representación del repositorio (figura 2.6) se encuentra en la clase “`saveContentSetupWizard.java`” ubicado en el paquete `beans`.
- El algoritmo para exportar el repositorio se encuentra en la clase “`exportToFolder.java`” ubicado en el paquete `save_restore`.
- El algoritmo para regular la cantidad de hilos a levantar por cada copia de seguridad se encuentra en la clase “`threadControl.java`” ubicado en el paquete “`save_restore`”.

Con la arquitectura utilizada en la implementación del AMP se garantiza que los algoritmos y funcionalidades propuestas se integren correctamente al cliente web Alfresco Explorer del ECM Alfresco Community 3.0.

2.3.2. Características de las funcionalidades del AMP propuesto

Para lograr un mejor entendimiento de los elementos que componen los paquetes y de la forma en que interactúan a continuación se brinda una explicación del funcionamiento general de la solución en base a las funcionalidades y paquetes antes mencionados:

Realizar copia de seguridad: La interfaz “admin-console.jsp”, redefinida en el paquete (alfrescoJsp), es el panel de administración de Alfresco. Desde esta página se accede a la opción “Realizar Copia de Seguridad” donde la clase “saveContentSetupWizard.java” (paquete beans) es la encargada de generar el wizard que guía al administrador paso a paso en la introducción de los datos necesarios para la ejecución de las copias de seguridad. El wizard genera las siguientes páginas ubicadas en el paquete (saveContent):

1. ci_what_to_save.jsp: página para escoger los contenidos, espacios y archivos de configuración a incluir en la copia de seguridad.
2. ci_save_local.jsp: página para escoger la ubicación donde desea guardar la copia.
3. ci_type_of_save.jsp: página para escoger si la copia a ejecutar será completa o incremental (actualmente el AMP sólo se permite las copias completas).
4. ci_summary.jsp: página que muestra el resumen de las opciones escogidas para realizar la copia de seguridad.
5. ci_message_log.jsp: página que muestra el resumen de los archivos exportados.

Las páginas incluyen botones que posibilitan la navegación entre ellas.

La clase “saveContentSetupWizard.java” utiliza al paquete (save_restore) donde se encuentran todas las clases con las funcionalidades necesarias para completar el proceso de realización de copias de seguridad. Como resultado de la copia se generan los ficheros “realizedBackup.bin” y “backupLogFile.txt” con datos sobre la misma.

Los paquetes (save_restore) y (beans) utilizan los servicios de Alfresco para visualizar el repositorio y exportarlo (ver anexo 2).

Programar copia de seguridad: Desde la interfaz “admin-console.jsp” del panel de administración de Alfresco se accede a la opción “Realizar Copia de Seguridad” donde la clase “saveContentSetupWizard.java” es la encargada de generar el wizard que guía paso a paso al administrador en la introducción de los datos necesarios para la programación de las copias de seguridad. El wizard genera las siguientes páginas:

1. ci_what_to_save.jsp: página para escoger los contenidos, espacios y archivos de configuración a incluir en la copia de seguridad.
2. ci_save_local.jsp: página para escoger la ubicación donde desea guardar la copia.
3. ci_type_of_save.jsp: página para escoger si la copia a ejecutar será completa o incremental (actualmente el AMP sólo se permite las copias completa).
4. ci_program_save.jsp: página que permite introducir la fecha, hora y frecuencia con que se ejecutará la copia de seguridad (paquete saveContent).
5. ci_summary.jsp: página que muestra el resumen de las opciones escogidas en la programación de la copia de seguridad.

Las páginas incluyen botones que posibilitan la navegación entre ellas.

La clase “saveContentSetupWizard.java” utiliza al paquete “save_restore” donde se encuentran las clases con las funcionalidades necesarias para la programación de copias de seguridad mediante en el cron de Alfresco. Como resultado de dicha acción se genera el fichero “programmedBackup.bin” con los datos sobre la copia programada que son necesarios a la hora de programar en el cron de Alfresco la ejecución de copia de seguridad cada vez que el mismo se reinicia, así como para la edición de dichas copias. El paquete (beans) utiliza los servicios de Alfresco para visualizar el repositorio (ver anexo 2).

Gestionar copias de seguridad realizadas: el administrador podrá ejecutar las acciones “Listar”, “Restaurar” y “Ver detalles”. Para ello debe acceder a la interfaz “admin-console.jsp” el cual es el panel de administración de Alfresco. Desde esta página se accede a la opción “Gestionar Copias de Seguridad Realizadas” donde la clase “restoreContentDialog.java” (paquete beans) se encargada de generar la página

“restoreContentDialog.jsp” (paquete restoreContent) quien muestra el listado de dichas copias.

El listado mostrado incluye las acciones “Ver detalles” y “Restaurar”, la primera acción la realiza la clase “showDetailsBackupDialog.java” (paquete beans) la cual genera la página “detailsBackupDialog.jsp” (paquete restoreContent) donde se muestra información (contenidos incluidos en la copia, etc.) referente a la copia de seguridad escogida.

La segunda acción se ejecuta mediante la clase “stopAndRestoreBackupWizard.java” (paquete beans) que mediante un wizard guía al administrador paso a paso en la introducción de datos para realizar la restaura de la copia de seguridad. El wizard genera las siguientes páginas ubicadas en el paquete (restoreContent):

1. ci_selectFilesDialog.jsp: página para seleccionar los contenidos, espacios y archivos de configuración a restaurar.
2. ci_selectOptionsDialog.jsp: página para seleccionar las opciones de restauración (espacio donde se ubicarán los documentos a importar, etc).
3. ci_restoreSummary.jsp: página que muestra el resumen de la restauración realizada.

Las páginas incluyen botones que posibilitan la navegación entre ellas.

Las clases “stopAndRestoreBackupWizard.java” y “showDetailsBackupDialog.java” utilizan al paquete (save_restore) el cual contiene las clases con las funcionalidades necesarias para acceder a los ficheros “realizedBackup.bin” y “backupLogFile.txt” con la información necesaria para realizar la restaura (ver anexo 2).

Gestionar copias de seguridad programadas: a través de esta funcionalidad el administrador podrá “Listar”, “Editar” y “Eliminar” las copias de seguridad programadas. Para ello debe acceder a la interfaz “admin-console.jsp” que es el panel de administración de Alfresco.

Luego de escoger la opción “Gestionar Copias de Seguridad Programadas” la clase “restoreContentDialog.java” es la encargada de mostrar el listado de dichas copias. El listado mostrado incluye las acciones “Eliminar” y “Editar”, la primera acción es

realizada por la clase “deleteProgrammedBackup.java” (paquete beans) la que genera la página “delete-program-save.jsp” (paquete restoreContent) donde se solicita que el administrador confirme la eliminación de la copia de seguridad escogida.

La segunda acción permite modificar la frecuencia de ejecución de la copia de seguridad programada a través de la página “editProgrammedBackup.jsp” (paquete restoreContent) generada por la clase “editProgrammedBackup.java” (paquete beans). Las clases “deleteProgrammedBackup.java” y “editProgrammedBackup.java” utilizan al paquete “save_restore” el cual contiene las funcionalidades para gestionar al fichero “programmedBackup.bin” que contiene la información sobre las copias programadas (ver anexo 2).

Conclusiones parciales

1. El algoritmo creado para recorrer y exportar los contenidos permite visitar una cantidad menor de espacios en comparación con los algoritmos tradicionales a la hora de exportar el repositorio.
2. El algoritmo creado para visualizar el repositorio elimina la necesidad de recorrerlo en su totalidad para su representación en el componente visual.
3. Con la implementación de un algoritmo que regula la cantidad de hilos a levantar por cada copia de seguridad en función del tamaño y cantidad de documentos, así como del tiempo de ejecución de las mismas se logra establecer una cantidad media de hilos a levantar.
4. El AMP creado permite agregar un conjunto de funcionalidades al cliente web Alfresco Explorer posibilitando visualizar el repositorio de Alfresco así como exportar los contenidos incluidos de manera selectiva en las copias de seguridad.

3. Valoración de la solución propuesta para la gestión de copias de seguridad desde cliente web Alfresco Explorer del ECM Alfresco Community 3.0

En el presente capítulo se realiza la validación de la solución propuesta por dos vías diferentes. Con el objetivo de validar la eficiencia se utiliza un método comparativo donde se mide tiempo de ejecución y espacio ocupado por las copias de seguridad desde la consola del SO y desde el módulo propuesto. Luego, utilizando la prueba “U de Mann-Whitney” [29] se demuestra que la solución propuesta presenta una mejora en la usabilidad con respecto a la variante brindada en la página oficial de Alfresco.

3.1. Valoración de la eficiencia en la gestión de las copias de seguridad de la solución propuesta

A continuación se muestran los resultados de la realización de varias copias de seguridad con el módulo propuesto y desde la consola del SO, utilizando la variante propuesta en la página oficial de Alfresco. Además se establece una comparación de la eficiencia de ambas variantes, en términos de tiempo de ejecución y espacio de almacenamiento.

De las copias realizadas se muestran los espacios a incluir, así como el tamaño que ocupan en el dispositivo de almacenamiento y el tiempo de ejecución. Las copias se realizan desde la consola de los SO Windows 8 y Debian GNU/Linux. El módulo propuesto puede incorporarse en el Alfresco Community 3.0 independientemente de la versión del SO Windows o GNU/Linux donde se encuentre ejecutándose.

Se realizarán un total de ocho copias de seguridad, seis desde Windows y dos desde GNU/Linux, cada una de las copias se ejecutará desde la consola del SO y desde el módulo propuesto. Las copias realizadas en Windows utilizan una computadora que tiene instalado un SO Windows 8 a 64 bit y cuenta con 4 gigabytes de memoria RAM⁵ y un microprocesador Intel Pentium Core i3 a 2.13 GHz, el ECM Alfresco utiliza el SGBD Derby.

Las copias ejecutadas desde GNU/Linux se realizan en una computadora que tiene instalado un SO Debian GNU/Linux con 1.5 gigabytes de memoria RAM y un

⁵ (Random Access Memory) es un tipo de memoria de computadora que puede ser accedida aleatoriamente (cada byte de memoria puede ser accedido sin necesidad de acceder al anterior).

microprocesador Intel Pentium a 2.13 GHz, el SGBD utilizado por Alfresco es PostgreSQL.

Para un repositorio de contenidos con 2.57 gigabytes.

Copia de seguridad 1: se realiza desde la consola del SO Windows.

Tamaño de la copia: 2.64 gigabytes (incluye todo el repositorio, índices de Lucene y backup de la Base de Datos).

Tiempo de ejecución: 6 minutos y 28 segundos.

Copia de seguridad 1: se realiza desde Windows utilizando el módulo propuesto.

A continuación se muestra la figura 3.1 con los espacios a seleccionar para exportar todo el repositorio:



Figura 3.1 Contenidos a incluir en la copia de seguridad.

Tamaño de la copia: 1.74 gigabytes

Tiempo de ejecución: 15 minutos

Para un repositorio de 2.57 gigabytes, el tiempo de ejecución de la copia de seguridad, de todo el repositorio, desde el módulo propuesto es superior a la variante desde la consola del SO. Por otro lado, el tamaño de la copia de seguridad desde el módulo propuesto es menor a la variante desde la consola del SO.

Copia de seguridad 2: exportando el espacio “/Company Home/User Homes/--- Proyecto” desde la consola del SO Windows.

Tamaño de la copia: 2.64 gigabytes (incluye todo el repositorio, índices de Lucene y backup de la Base de Datos).

Tiempo de ejecución: 6 minutos.

Copia de seguridad 2: exportando el espacio “/Company Home/User Homes/---- Proyecto” desde Windows utilizando el módulo propuesto.
A continuación se muestra la figura 3.2 del espacio a incluir:



Figura 3.2 Contenidos a incluir en la copia de seguridad.

Tamaño de la copia: 137 megabytes.

Tiempo de ejecución: 3 minutos.

Para un repositorio de 2.57 gigabytes, el tiempo de ejecución y el espacio ocupado por la copia de seguridad de la carpeta “/Company Home/User Homes/----Proyecto” desde el módulo propuesto son inferiores a la variante desde la consola del SO.

Copia de seguridad 3: exportando el documento “/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc” desde la consola del SO Windows.

Tamaño de la copia: 2.64 gigabytes (incluye todo el repositorio, índices de Lucene y backup de la Base de Datos).

Tiempo de ejecución: 6 minutos.

Copia de seguridad 3: exportando el documento “/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc” desde Windows utilizando el módulo propuesto.

A continuación se muestra la figura 3.3 con el documento a incluir:



Figura 3.3 Contenidos a incluir en la copia de seguridad.

Tamaño de la copia: 64.0 kilobytes

Tiempo de ejecución: no llega al segundo

Para un repositorio de 2.57 gigabytes, el tiempo de ejecución y el espacio ocupado por la copia de seguridad del documento “/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc” desde el módulo propuesto son muy inferiores a la variante desde la consola del SO.

Para un repositorio de contenidos con 3.52 gigabytes

Copia de seguridad 4: exportando todo el repositorio desde la consola del SO Windows.

Tamaño de la copia: 3.68 gigabytes (incluye al repositorio, índices de Lucene y backup de la Base de Datos).

Tiempo de ejecución: 8 minutos.

Copia de seguridad 4: exportando todo el repositorio utilizando el módulo propuesto desde Windows.

(Ver figura 3.1)

Tamaño de la copia: 2.47 gigabytes.

Tiempo de ejecución: 22 minutos.

Para un repositorio de 3.52 gigabytes, el tiempo de ejecución de la copia de seguridad, de todo el repositorio, desde el módulo propuesto es superior a la variante desde la consola del SO. Por otro lado, el tamaño de la copia de seguridad desde el módulo propuesto es menor a la variante desde la consola del SO.

Copia de seguridad 5: exportando el espacio “/Company Home/User Homes/Tutoriales” desde la consola del SO Windows.

Tamaño de la copia: 3.68 gigabytes (incluye al repositorio, índices de Lucene y backup de la Base de Datos).

Tiempo de ejecución: 8 minutos.

Copia de seguridad 5: exportando el espacio “/Company Home/User Homes/Tutoriales” utilizando el módulo propuesto desde Windows.

A continuación se muestra la figura 3.4 con el espacio a incluir:



Figura 3.4 Contenidos a incluir en la copia de seguridad.

Tamaño de la copia: 803 megabytes.

Tiempo de ejecución: 7 minutos.

Para un repositorio de 3.52 gigabytes, el tiempo de ejecución y el espacio ocupado por la copia de seguridad de la carpeta “/Company Home/User Homes/Tutoriales” desde el módulo propuesto son inferiores a la variante desde la consola del SO.

Copia de seguridad 6: exportando el documento “/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc” desde la consola del SO de Windows.

Tamaño de la copia: 3.68 gigabytes (incluye al repositorio, índices de Lucene y backup de la Base de Datos).

Tiempo de ejecución: 8 minutos.

Copia de seguridad 6: exportando el documento “/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc” utilizando el módulo propuesto desde Windows.

A continuación se muestra la figura 3.5 con el documento a incluir:

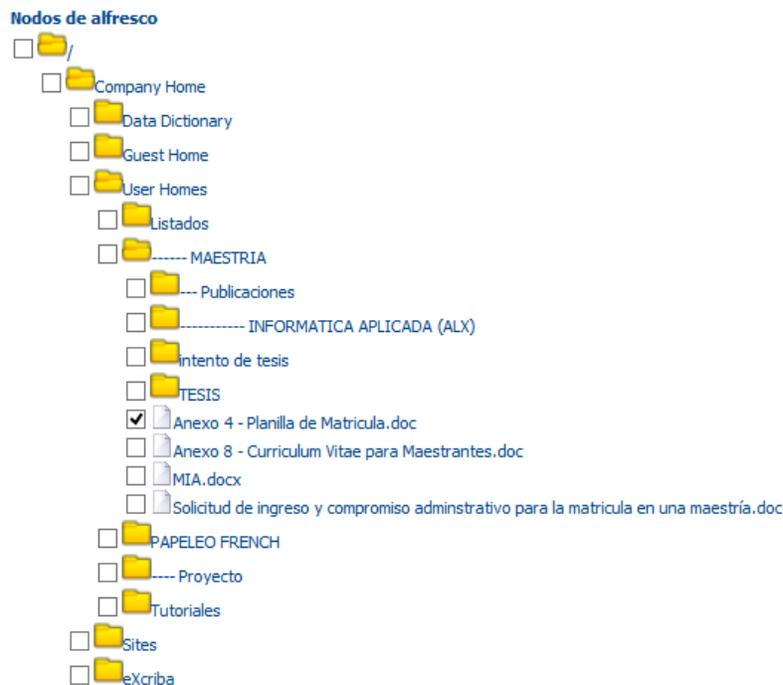


Figura 3.5 Contenidos a incluir en la copia de seguridad.

Tamaño: 2.73 megabytes

Tiempo de ejecución: 3 segundos

Para un repositorio de 3.52 gigabytes, el tiempo de ejecución y el espacio ocupado por la copia de seguridad del documento “/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc” desde el módulo propuesto son muy inferiores a la variante desde la consola del SO.

Para un repositorio de contenidos con 1.1 gigabytes

Copia de Seguridad 7: exportando todo el repositorio desde la consola del SO Debian GNU/Linux.

Tamaño de la copia: 1.1 gigabytes (incluye todo el repositorio, índices de Lucene y backup de la Base de Datos de PostgreSQL)

Tiempo de ejecución: 3 minutos

Copia de Seguridad 7: exportando todo el repositorio utilizando el módulo propuesto desde Debian GNU/Linux.

A continuación se muestra la figura 3.6 con los espacios a seleccionar para exportar el repositorio:



Figura 3.6 Contenidos a incluir en la copia de seguridad.

Tamaño de la copia: 614.2 megabytes

Tiempo de ejecución: 8 minutos

Para un repositorio de 1.1 gigabytes, el tiempo de ejecución de la copia de seguridad, de todo el repositorio, desde el módulo propuesto es superior a la variante desde la consola del SO. Por otro lado, el tamaño de la copia de seguridad desde el módulo propuesto es menor a la variante desde la consola del SO.

Copia de Seguridad 8: exportando el espacio “/Company Home/User Homes/Gestion Documental” desde la consola del SO Debian GNU/Linux.

Tamaño de la copia: 1.1 gigabytes (incluye todo el repositorio, índices de Lucene y backup de la Base de Datos de PostgreSQL).

Tiempo de ejecución: 3 minutos.

Copia de Seguridad 8: exportando el espacio “/Company Home/User Homes/Gestion Documental” utilizando el módulo propuesto desde GNU/Linux.

A continuación se muestra la figura 3.7 con el espacio a exportar:



Figura 3.7 Contenidos a incluir en la copia de seguridad.

Tamaño de la copia: 2.0 megabytes.

Tiempo de ejecución: 2 segundos.

Para un repositorio de 1.1 gigabytes, el tiempo de ejecución y el espacio ocupado por la copia de seguridad de la carpeta “/Company Home/User Homes/Gestion Documental” desde el módulo propuesto son muy inferiores a la variante desde la consola del SO.

3.1.1. Análisis del espacio de almacenamiento

En el presente epígrafe se muestra un resumen del espacio que ocupan las copias de seguridad antes mostradas y se realiza un análisis de las mismas en base al espacio que ocupan en el dispositivo de almacenamiento lo cual es mostrado en la tabla 3.1:

Tabla 3.1 Espacio que ocupan las copias de seguridad en el dispositivo de almacenamiento.

Para un repositorio con 2.57 gigabytes (Windows 8)			
Copia 1	Consola	/Company Home	2.64 gigabytes
	AMP	/Company Home	1.74 gigabytes
Espacio ahorrado por el AMP			900 megabytes
Copia 2	Consola	/Company Home/User Homes/----Proyecto	2.64 gigabytes
	AMP	/Company Home/User Homes/----Proyecto	137 megabytes
Espacio ahorrado por el AMP			2.4 gigabytes
Copia 3	Consola	/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc	2.64 gigabytes
	AMP	/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc	64 kilobytes
Espacio ahorrado por el AMP			2.64 gigabytes
Para un repositorio con 3.52 gigabytes (Windows 8)			
Copia 4	Consola	/Company Home	3.68 gigabytes
	AMP	/Company Home	2.47 gigabytes
Espacio ahorrado por el AMP			1.21 gigabytes
Copia 5	Consola	/Company Home/User Homes/Tutoriales	3.68 gigabytes
	AMP	/Company Home/User Homes/Tutoriales	803 megabytes
Espacio ahorrado por el AMP			2.87 gigabytes
Copia 6	Consola	/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc	3.68 gigabytes
	AMP	/Company Home/User Homes/-----	2.73

		Maestría/Anexo 4 - Planilla de Matricula.doc	megabytes
Espacio ahorrado por el AMP			3.67 gigabytes
Para un repositorio con 1.1 gigabytes (Debian)			
Copia 7	Consola	/Company Home	1.1 gigabyte
	AMP	/Company Home	614.2 megabytes
Espacio ahorrado por el AMP			485.8 megabytes
Copia 8	Consola	/Company Home/User Homes/Gestion Documental	1.1 gigabytes
	AMP	/Company Home/User Homes/Gestion Documental	2.0 megabytes
Espacio ahorrado por el AMP			1.09 gigabytes

De la tabla 3.1 se tiene que la columna 1 contiene el número de la copia de seguridad realizada, la columna 2 contiene desde dónde se realizó la copia de seguridad (consola del SO o el módulo propuesto), la columna 3 contiene los documentos a incluir en la copia de seguridad y la columna 4 contiene el espacio que ocupa la copia en el dispositivo de almacenamiento. Las filas que muestran el espacio ahorrado por el módulo indican el ahorro de espacio que representa realizar la copia de seguridad desde el AMP propuesto con respecto a la copia realizada desde el SO.

Como resultado de la comparación realizada, en base al espacio que ocupan las copias de seguridad, se verifica que desde la consola del SO, independientemente de los documentos que se incluyan y del SO, el tamaño de la copia siempre es superior al del repositorio. Ello sucede debido a que la copia de seguridad, desde la consola del SO, incluye los backup del repositorio, índices de Lucene y la BD; los dos últimos provocan un aumento de su tamaño total. Ejemplo de lo anterior son las copias Copia 1 y Copia 4 que, realizadas desde la consola del SO, ocupan un tamaño superior al repositorio como resultado de agregarles los backup de los índices de Lucene y la BD.

En comparación a las copias que se realizan desde la consola del SO, el módulo propuesto permite seleccionar, desde el cliente web Alfresco Explorer, los documentos a incluir en las copias de seguridad. Ello permite que el tamaño total de las copias sea igual al del total de documentos incluidos en las mismas. Por otro lado, desde la consola del SO, siempre es necesario incluir todo el repositorio, independientemente de los documentos que realmente se deseen mantener en la copia.

Además del ahorro de espacio que implica el poder seleccionar los documentos a incluir en la copia de seguridad, el módulo permite optimizar espacio debido a que los contenidos se exportan como ACP, los cuales son archivos comprimidos. Ello se evidencia cuando en la Copia 6 realizada utilizando el módulo propuesto el documento “Alfresco-Tutorial.doc” tiene un tamaño de 4.2 megabytes, sin embargo, cuando se exporta por el AMP, como un ACP, ocupa un espacio de 2.7 megabytes, casi la mitad.

Lo mencionado en el párrafo anterior explica el por qué cuando se exporta todo el repositorio desde el AMP propuesto (ver Copia 1) se utilizan 1.74 gigabytes de espacio y no 2.64 gigabytes (tamaño del repositorio). La Copia 4 utiliza 2.47 gigabytes en lugar de 3.68 gigabytes (tamaño del repositorio) y la Copia 7 utiliza 485.8 megabytes en lugar de 1.1 gigabytes (tamaño del repositorio).

Luego del análisis realizado con respecto al espacio de almacenamiento ocupado por las copias de seguridad, con el módulo propuesto y con la variante ejecutada desde la consola del SO, se tiene que el espacio ocupado es superior en esta última lo que influye en que el módulo propuesto tenga una mayor eficiencia. Ello incide directamente en el ahorro de espacio en los dispositivos de almacenamiento de las entidades o empresas donde este sea desplegado.

3.1.2. Análisis del tiempo de ejecución

En el presente epígrafe se muestra el tiempo que demoran las copias de seguridad en su ejecución para luego realizar un análisis de los mismos y mostrar cómo repercute en la eficiencia. La tabla 3.2 muestra un resumen los resultados obtenidos con respecto al tiempo de ejecución de las copias:

Tabla 3.2 Tiempo de ejecución de las copias de seguridad.

Para un repositorio con 2.57 gigabytes (Windows 8)			
Copia 1	Consola	/Company Home	6 minutos y 28 segundos
	AMP	/Company Home	15 minutos
Tiempo ahorrado por el AMP			-----
Copia 2	Consola	/Company Home/User Homes/----Proyecto	6 minutos
	AMP	/Company Home/User Homes/----Proyecto	3 minutos
Tiempo ahorrado por el AMP			3 minutos
Copia 3	Consola	/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc	6 minutos
	AMP	/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc	pocos segundos
Tiempo ahorrado por el AMP			6 minutos
Para un repositorio con 3.52 gigabytes (Windows 8)			
Copia 4	Consola	/Company Home	8 minutos
	AMP	/Company Home	22 minutos
Tiempo ahorrado por el AMP			-----
Copia 5	Consola	/Company Home/User Homes/Tutoriales	8 minutos
	AMP	/Company Home/User Homes/Tutoriales	7 minutos
Tiempo ahorrado por el AMP			1 minuto
Copia 6	Consola	/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc	8 minutos
	AMP	/Company Home/User Homes/----- Maestría/Anexo 4 - Planilla de Matricula.doc	pocos segundos
Tiempo ahorrado por el AMP			8 minutos
Para un repositorio con 1.1 gigabytes (Debian)			

Copia 7	Consola	/Company Home	3 minutos
	AMP	/Company Home	8 minutos
Tiempo ahorrado por el AMP			-----
Copia 8	Consola	/Company Home/User Homes/Gestion Documental	3 minutos
	AMP	/Company Home/User Homes/Gestion Documental	pocos segundos
Tiempo ahorrado por el AMP			3 minutos

En la tabla 3.2 se observa que en comparación con el tiempo de ejecución de la variante propuesta en la página oficial de Alfresco (Consola), para cierta cantidad de documentos, el tiempo de ejecución del módulo propuesto es menor. Ello no sucede para todos los volúmenes de documentos debido a que la solución propuesta siempre crea un archivo comprimido por cada documento a exportar lo que aumenta el tiempo de ejecución de la exportación. No obstante, el tiempo de ejecución de las copias de seguridad, ejecutadas desde el módulo, está acorde al tamaño total del volumen de documentos incluidos, mientras mayor sea el tamaño total de documentos mayor es el tiempo de ejecución y viceversa. Para una determinada cantidad de documentos la eficiencia de la solución propuesta es superior a la de la variante desde la consola del SO.

De los análisis realizados con respecto al tiempo de ejecución y al espacio ocupado por las copias de seguridad se llega a la conclusión que la utilización de algoritmos para la visualización y exportación de los contenidos hacen posible que el espacio ocupado por las copias y su tiempo de ejecución sea proporcional a la cantidad de documentos que se incluyan en las mismas, lo cual provoca una mejora de la eficiencia acorde a la definición dada en el marco del presente trabajo de investigación.

3.2. Validación de la usabilidad

Según [30] usabilidad es un atributo de calidad que mide que tan fácil de utilizar es una interfaz de usuario. Su definición formal es resultado de la enumeración de los diferentes componentes o variables a través de los cuales puede ser medida. Entre estos componentes se encuentran:

- Facilidad de Aprendizaje: ¿Cómo de fácil resulta para los usuarios llevar a cabo tareas básicas la primera vez que se enfrentan al diseño?
- Eficiencia: Una vez que los usuarios han aprendido el funcionamiento básico del diseño, ¿cuánto tardan en la realización de tareas?
- Calidad de ser recordado: Cuando los usuarios vuelven a usar el diseño después de un periodo sin hacerlo, ¿cuánto tardan en volver a adquirir el conocimiento necesario para usarlo eficientemente?
- Eficacia: Durante la realización de una tarea, ¿cuántos errores comete el usuario?, ¿cómo de graves son las consecuencias de esos errores?, ¿cómo de rápido puede el usuario deshacer las consecuencias de sus propios errores?
- Satisfacción: ¿Cómo de agradable y sencillo le ha parecido al usuario la realización de las tareas?

Teniendo en cuenta los atributos antes mencionados se realiza una encuesta (ver Anexo 1) donde se mide la usabilidad del AMP propuesto y la de la variante, propuesta en la página oficial del ECM Alfresco, ejecutada desde la consola del SO. Los quince encuestados son posibles usuarios finales de la solución propuesta y cuentan con experiencia en el trabajo con el SO GNU/Linux y con sistemas de gestión documental.

Para determinar cuál variante presenta mayor usabilidad se realizará la Prueba “U de Mann-Whitney”, la cual es una prueba estadística no paramétrica que presenta las siguientes características:

- Es libre de curvas, no necesita una distribución específica.
- Nivel ordinal de la variante dependiente.
- Se utiliza para comparar dos grupos de rangos (medianas) y determinar que la diferencia no se deba al azar (que la diferencia sea estadísticamente significativa) [29].

En este caso los grupos a comparar son “Módulo propuesto” y “Consola”. La variable dependiente en torno a la cual se realizará la comparación es “Usabilidad”.

Para la realización de la prueba se presenta la siguiente hipótesis alterna:

Ha: el módulo propuesto, acorde a las calificaciones obtenidas en la encuesta, presenta mayor usabilidad que la variante propuesta en la página oficial de Alfresco.

Con la hipótesis nula **H₀**: las diferencias entre las calificaciones obtenidas en la encuesta, con respecto a la usabilidad, no son significativas.

Para todo valor de probabilidad igual o menor que $p = 0.05$ se acepta **H_a** y rechaza **H₀**, para todo valor de probabilidad mayor de $p = 0.05$ se acepta **H₀** y se rechaza **H_a**.

Por cada una de las propuestas “Consola” y “Módulo” se recoge el total de puntuación obtenido en base a cada uno de los componentes antes mencionados para medir la usabilidad:

Tabla 3.2 Total de puntuación obtenido por cada componente.

	Sobre la facilidad de aprendizaje y la eficiencia		Respecto a la eficacia			Satisfacción
	P1	P2	P1	P2	P3	P1
Módulo	72	70	69	67	65	71
Consola	42	50	49	50	53	36

En la tabla de la figura 3.2 se muestra que de los cinco componentes por los que es posible medir la usabilidad, se utilizaron cuatro para elaborar la encuesta, ellos son: facilidad de aprendizaje, eficiencia, eficacia y satisfacción, debido a que el componente restante “Calidad de ser recordado” no se tuvo en cuenta como resultado de la disponibilidad de manuales o tutoriales para cada una de las variantes de ejecución de copias de seguridad. **P1**, **P2** y **P3** hacen referencia a las preguntas que aparecen en la encuesta por cada uno de los componentes antes mencionados.

Como herramienta de apoyo para la realización de la prueba se utiliza el SPSS (versión 2013) al cual se le introducen los valores resumidos en la tabla anterior. Luego de aplicar a dichos valores la prueba “U de Mann-Whitney” la herramienta arroja los siguientes resultados tomados a partir de una imagen mostrada por la misma:

→ NPar Tests

Mann-Whitney Test

	Variante	N	Mean Rank	Sum of Ranks
Valor	1	6	9.50	57.00
	2	6	3.50	21.00
	Total	12		

	Valor
Mann-Whitney U	.000
Wilcoxon W	21.000
Z	-2.887
Asymp. Sig. (2-tailed)	.004
Exact Sig. [2*(1-tailed Sig.)]	.002 ^a

a. Not corrected for ties.

b. Grouping Variable: Variante

Figura 3.8 Resultado de la prueba “U de Mann-Whitney”.

En la figura 3.8 se muestran los resultados brindados por SPSS. Ranks presenta una descripción de los grupos comparados respecto a la media y suma de rangos.

En Test Statistics se presentan los valores de:

- U de Mann Whitney: 0.
- Razón Z: -2.887.
- Nivel de significancia de la prueba: 0.004.

Como en este caso el nivel de significancia es menor o igual que 0.05 se rechaza la hipótesis nula y acepta la hipótesis alternativa, por lo que el módulo propuesto presenta mayor usabilidad que la variante que aparece en la página oficial de Alfresco.

Conclusiones parciales

1. El resultado de la encuesta aplicada en base a los componentes por los cuales se mide la usabilidad arrojó que el módulo propuesto presenta mayor usabilidad que la variante ejecutada desde la consola del SO.
2. La comparación realizada para medir la eficiencia arrojó como resultado que la del módulo propuesto es mayor debido a que disminuye el espacio que ocupan las copias de seguridad.
3. Otro factor arrojado por el método de comparación aplicado es que el tiempo de ejecución de las copias de seguridad incide en la mejor eficiencia del módulo propuesto puesto que, para una determinada cantidad de documentos, es menor al de la variante ejecutada desde la consola del SO.

Conclusiones

Después de realizar la presente investigación se concluye:

- Los ECM open source analizados no incluyen desde sus clientes web funcionalidades para visualizar, seleccionar y exportar los contenidos de sus repositorios. Para realizarles copias de seguridad a sus repositorios, obligatoriamente, se necesita almacenar todos los contenidos existentes en los mismos, ello conlleva a ineficiencias en lo que a espacio de almacenamiento y tiempo de ejecución respecta.
- Al visualizar el repositorio de Alfresco utilizando una estructura de tipo árbol, a través del algoritmo empleado para ello, es posible recorrer dicha estructura utilizando una variación del algoritmo en preorden que visita únicamente los nodos seleccionados, disminuyendo así el tiempo de procesamiento del algoritmo.
- Con la implementación de un algoritmo que permite regular la cantidad de hilos a levantar se pueden realizar copias de seguridad en paralelo sin que afecte el rendimiento del ECM Alfresco.
- Al incluir de manera selectiva, desde el cliente web Alfresco Explorer, los contenidos a exportar en la copia de seguridad se logra disminuir el espacio y tiempo de ejecución de las mismas para una determinada cantidad de contenidos.
- A partir de los algoritmos desarrollados se creó un módulo que al ser incorporado al cliente web Alfresco Explorer del Alfresco Community 3.0 elimina la necesidad de utilizar las herramientas para la ejecución de copias de seguridad en sistemas de archivos y bases de datos desde la consola del SO, lo cual permite que usuarios con conocimientos mínimos puedan gestionar las copias de seguridad del ECM Alfresco Community 3.0.
- Con la utilización del método comparativo, así como de la prueba “U de Mann-Whitney” se logró comprobar que el módulo propuesto es más eficiente y tiene mayor usabilidad que la variante propuesta desde la consola del SO en la página oficial de Alfresco.

Recomendaciones

- Valorar la implementación de un algoritmo que permita almacenar los archivos exportados en una estructura de directorios semejantes a la que presenta en el repositorio de Alfresco.
- Valorar la implementación de un nuevo mecanismo para visualizar el árbol que muestra los contenidos a importar.

Referencias Bibliográficas

- [1] MENA, Mayra. *Gestión documental y organización de los archivos*. La Habana: Félix Varela, 2005. ISBN: 959-258-950-X.
- [2] Documentum FAQ. [en línea] [ref. de 12 de junio 2012]; Disponible en Web: <http://www.ohelp.com/samples/tutorial/documentum-FAQ.pdf>.
- [3] Openkm. [en línea] [Ref. 12 junio 2012]; Disponible en Web: <http://www.openkm.com/es/>.
- [4] MUNWAR, Shariff. *Alfresco Enterprise Content Management Implementation*. S.l.: Packt Publishing, 2006. ISBN: 1-904811-11-6.
- [5] EUROPEAN COMMISSION. *Moreq2 SPECIFICATION*. Francia: Luxembourg: Office for Official Publications of the European Communities, 2008. ISBN:978-92-79-09772-0.
- [6] ALFRESCO INC. Alfresco docs. [en línea] [ref. de 11 de mayo 2013]; Disponible en Web: http://docs.alfresco.com/3.4/topic/com.alfresco.Enterprise_3_4_0.doc/concepts/ch-backup-restore.html.
- [7] Jconsole. [en línea] [ref. de 11 de mayo 2013]; Disponible en Web: <http://openjdk.java.net/tools/svc/jconsole/>.
- [8] HEILEMAN, Gregory L. *Estructura de datos, algoritmos y programación orientada a objetos*. La Habana: Félix Varela, 2003.
- [9] Tipos de copia de seguridad. [en línea] [ref. de 13 de junio 2012]; Disponible en Web: http://www.ite.educacion.es/formacion/materiales/85/cd/REDES_LINUX/backup/Ti_pos_de_copia_de_seguridad.html.
- [10] AIIM. [en línea] [ref. de 9 de julio 2012]; Disponible en Web: <http://www.aiim.org/What-is-ECM-Enterprise-Content-Management>.
- [11] Openkm repository backup. [en línea] [ref. de 6 de junio 2012]; Disponible en Web: http://wiki.openkm.com/index.php/Repository_backup.
- [12] Nuxeo backups. [en línea] [ref. de 7 de junio 2012]; Disponible en Web: <http://doc.nuxeo.com/display/ADMINDOC/Backup+and+restore>.
- [13] KNOWLEDGETREE INC. *Backing up and restoring*. [en línea] [ref. de 12 de junio 2012]; Disponible en Web: http://docs.knowledgetree.com/manuals/ig/backing_up_and_restoring.html.

- [14] Magnolia backups. [en línea] [ref. de 12 de junio 2012]; Disponible en Web:<http://documentation.magnolia-cms.com/modules/backup.html>.
- [15] Intecna Soluciones. [en línea] [ref. de 15 de junio 2012]; Disponible en Web:<http://www.intecna-sistemas.es/>.
- [16] Junta de Andalucía. [en línea] [ref. de 15 de junio 2012]; Disponible en Web: http://www.juntadeandalucia.es/repositorio/usuario/peticiones/directaInfoBasica.jsf?linkDummyForm:_idcl=items_descarga:3:items_descarga:0:_id190&.
- [17] INTECNA SOLUCIONES. *Sistema de copias de seguridad flexible Manual de usuario* [en línea]. 2010. S.l.: s.n. [ref. de 15 de junio 2012]; Disponible en Web: http://www.juntadeandalucia.es/repositorio/usuario/peticiones/directa.jsf?linkDummyForm:_idcl=items_descarga:3:items_descarga:0:items_descarga:0:_id198&idArchivo=1510&.
- [18] AHO, Alfred V., HOPCROFT, John E. y ULLMAN, Jeffrey D. *Estructuras de datos y Algoritmos*. México: s.n., 1988. ISBN: 968-6048-19-7.
- [19] LAFORE, Robert. *Data Structures & Algorithms in Java*. S.l.: s.n., 1998. ISBN: 1571690956.
- [20] DEPARTAMENTO DE TÉCNICAS DE PROGRAMACIÓN. *Material de apoyo semana 8 Programación II*. 2011.
- [21] JACOBSON, I., BOOCH, G., y RUMBAUGH, J. *El proceso Unificado de Desarrollo de Software*. La Habana: Félix Varela, 2004.
- [22] LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. 2004. Vol. 1.
- [23] Increase productivity and enhance communication and collaboration efficiency by using UML. [En línea] [Citado el: 12 de julio de 2009.] <http://www.visual-paradigm.com>.
- [24] HOLZNER, Steven. *La Biblia de Java 2*. S.l.: Anaya Multimedia, [sin fecha].
- [25] Mozilla Developer Network. [en línea] [ref. de 27 de agosto 2012]; Disponible en Web: https://developer.mozilla.org/es/docs/Guía_JavaScript_1.5/Concepto_de_JavaScript.
- [26] ANDREWS, Gregory R. *Concurrent Programming: Principles and Practice*. S.l.: Addison-Wesley, 1991. ISBN: 978-0805300864.
- [27] ZUKOWSKI, John. *Java 2 J2SE 1.4*. España: Anaya Multimedia, 2003. ISBN: 84-415-1559-X.
- [28] POTTS, Jeff. *Alfresco Developer Guide*. S.l.: Packt Publishing, 2009.

- [29] ÁLVAREZ, Rafael. *ESTADÍSTICA APLICADA A LAS CIENCIAS DE LA SALUD*. S.l.: Díaz de Santos, 2007. ISBN 978-84-7978-823-0.
- [30] NIELSEN NORMAN GROUP. [ref. de 11 de mayo 2013]; Disponible en Web <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>.

Bibliografía

AMP Files - alfresco wiki. [en línea] [ref. de 30 de Junio 2012]; Disponible en Web: https://wiki.alfresco.com/wiki/AMP_Files.

Apache Derby. [en línea] [ref. de 30 Junio de 2012]; Disponible en Web: <http://db.apache.org/derby/>.

Bacula® - *La solución de Backup Open Source basada en red*. [en línea]. [ref. de 21 de Junio 2012]; Disponible en Web: <http://www.bacula.org/es/>.

BOOCH, G., RUMBAUGH, J. y JACOBSON, I. *The Unified Modeling Language User Guide*. 2da ed. S.I.: Addison-Wesley, 2005. ISBN: 978-0321267979.

BUBOIS, Paul, 2009. *MySQL Developer's Library*. 4ta. ed. S.I.: Pearson Education, Inc. ISBN 978-0-672-32938-8.

ECKEL, Bruce, 2003. *Thinking in Java*. 3ra. ed. S.I.: Pearson Education, Inc. ISBN 0-13-100287-2.

Fwbackups-Overview. [en línea] [ref. de 26 de Junio 2012]; Disponible en Web: <http://www.diffingo.com/oss/fwbackups>.

GOODMAN, Danny y MORRISON, Michael, 2004. *JavaScript Bible*. 5ta. ed. Indianapolis: Wiley Publishing, Inc. ISBN 0-7645-5743-2.

Grsync.[en línea] [ref. de 29 de Junio 2012]; Disponible en Web:<http://www.opbyte.it/grsync/>.

GRAU, Ricardo, CORREA, Cecilia y ROJAS, Mauricio, 2004. *Metodología de la Investigación*. 2da. ed. Ibagué -Tolima, Colombia: Fondo Editorial Coruniversitaria. ISBN 958-8028-10-8.

KERNIGHAN, Brian W. y RITCHIE, Dennis M. *The C Programming Language*. Second Edition. S.I.: Prentice Hall PTR, 1988. ISBN: 978-0131103627.

HERNÁNDEZ, Rolando A. y COELLO, Sayda, 2011. *El proceso de investigación científica*. S.I.: Editorial Universitaria. ISBN 978-959-16- 1307-3.

HSQLDB-100% Java Database. [en línea] [ref. 30 de 30 Junio 2012]; Disponible en Web: <http://www.hsqldb.org/>.

LARMAN, Craig, 1999. *UML y Patrones. Una introducción al análisis y diseño orientado a objeto y al proceso unificado*. México: Prentice Hall. ISBN 970-17-0261-1.

LUIS COBIAN, CobianSoft. [en línea] [ref. de 21 de Junio 2012]; Disponible en Web: <http://www.cobiansoft.com/cobianbackup.htm>.

PATZER, Andrew, 2003. *JSP Ejemplos prácticos*. España: ANAYA MULTIMEDIA. ISBN 84-415-1466-6.

Rsync. [en línea] [ref. de 28 de Junio 2012]; Disponible en Web: <http://rsync.samba.org/ftp/rsync/rsync.html>.

Scheduled Actions - alfresco wiki. [en línea] [ref. de 30 de Junio 2012]; Disponible en Web: http://wiki.alfresco.com/wiki/Scheduled_Actions.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2008. *PostgreSQL 8.3.14 Documentation*. 2008. S.l.: s.n.

WALLS, Craig y BREIDENBANCH, Ryan, 2008. *Spring In Action*. 2da. ed. S.l.: Manning Publications Co. ISBN 1-933988-13-4.

Anexos

Anexo 1: Encuesta Aplicada para medir la usabilidad

Hipótesis de la investigación: Con la implementación de algoritmos de programación que permitan la visualización y exportación del repositorio del ECM Alfresco Community 3.0, desde el cliente web Alfresco Explorer, mejorará la eficiencia y usabilidad en la gestión de sus copias de seguridad.

El presente cuestionario forma parte de una consulta para la valoración de un módulo que gestiona las de copias de seguridad desde el cliente web Alfresco Explorer del ECM Alfresco Community 3.0.

Con este fin se solicita afectuosamente su colaboración, teniendo en cuenta que sus opiniones serán de gran valor en este trabajo de investigación para validar o rectificar la propuesta, garantizando en todo el proceso la confidencialidad de sus respuestas, utilizando los datos únicamente con intenciones académico-científicas.

1. Datos personales

Nombre y Apellidos:

Centro de trabajo o estudio:

Grado científico:

Categoría Docente:

Cargo:

Nivel de enseñanza:

Especialidad:

Años de experiencia:

2. Marque con una (x), en la casilla que corresponda al grado de conocimientos que tiene en la gestión de copias de seguridad en sistemas de archivos y bases de datos. Valorándolo en una escala de 0 a 10 (considerando 0 como no tener absolutamente ningún conocimiento y 10 el de pleno conocimiento del tema).

0	1	2	3	4	5	6	7	8	9	10

3. Le pedimos que responda el siguiente cuestionario para obtener su valoración como usuario final del módulo propuesto.

Utilice los siguientes pesos (2 a 5) para evaluar los diferentes aspectos que se proponen.

Sobre la facilidad de aprendizaje y la eficiencia:

1. ¿Le resultó fácil realizar una copia de seguridad la primera vez?

5	4	3	2
Demasiado	Bastante	Poco	Muy poco

a. Desde la consola: 5____ 4____ 3____ 2____

b. Desde el módulo propuesto: 5____ 4____ 3____ 2____

2. Luego de aprender y estar familiarizado con la ejecución de copias de seguridad desde el módulo y la consola, ¿qué tiempo le toma la realización de las mismas?

2	3	4	5
Demasiado	Bastante	Poco	Muy poco

a. Desde la consola: 5____ 4____ 3____ 2____

b. Desde el módulo propuesto: 5____ 4____ 3____ 2____

Respecto a la eficacia:

1. ¿Durante la realización de las copias comete errores?

2	3	4	5
----------	----------	----------	----------

Demasiado	Bastante	Poco	Muy poco
-----------	----------	------	----------

a. Desde la consola: 5____ 4____ 3____ 2____

b. Desde el módulo propuesto: 5____ 4____ 3____ 2____

2. ¿Fueron graves las consecuencias de los errores cometidos?

2	3	4	5
Demasiado	Bastante	Poco	Muy poco

a. Desde la consola: 5____ 4____ 3____ 2____

b. Desde el módulo propuesto: 5____ 4____ 3____ 2____

3. ¿Le fue difícil deshacer los errores cometidos rápidamente?

2	3	4	5
Demasiado	Bastante	Poco	Muy poco

a. Desde la consola: 5____ 4____ 3____ 2____

b. Desde el módulo propuesto: 5____ 4____ 3____ 2____

Sobre la satisfacción:

1. ¿Le ha parecido agradable e intuitivo la ejecución de copias de seguridad?

5	4	3	2
Demasiado	Bastante	Poco	Muy poco

a. Desde la consola: 5____ 4____ 3____ 2____

b. Desde el módulo propuesto: 5____ 4____ 3____ 2_____

Anexo 2: Vistas del AMP propuesto

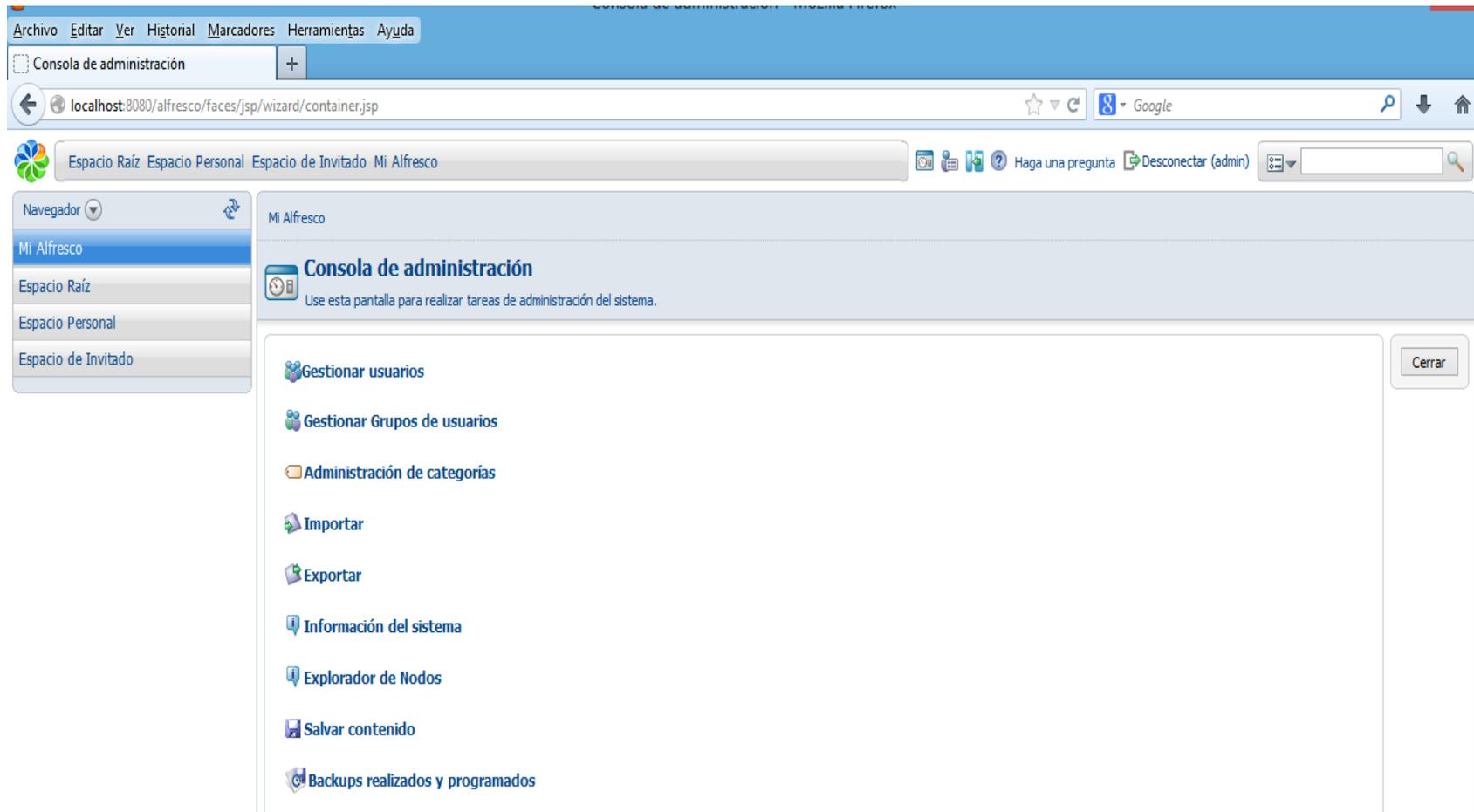


Figura 1 Panel de control del Alfresco Explorer.

The screenshot shows a web browser window with the URL `localhost:8080/alfresco/faces/jsp/wizard/container.jsp`. The page title is "Configuración para salvar el contenido". The navigation bar includes "Espacio Raíz", "Espacio Personal", "Espacio de Invitado", and "Mi Alfresco". A sidebar on the left shows a "Navegador" menu with "Mi Alfresco" selected. The main content area is titled "Configuración para salvar el contenido" and includes a sub-header "El wizard lo guiará paso a paso a través de las opciones necesarias para poder crear un backup".

Pasos

1. Qué incluir en el backup
2. Ubicación del backup
3. Crear un backup programado
4. Resumen
5. Información del archivo log

Selección de nodos para incluir en el backup

En esta página puede seleccionar que incluir en el backup que se encuentra realizando

Nodos de alfresco

- /
- Company Home
 - Data Dictionary
 - Guest Home
 - User Homes
- Sites
- eXcriba
 - Configurations
 - Workspace
 - Incoming
 - Templates
 - CC
 - Transforming
 - Fond

Para continuar pulse Siguiente.

Navigation buttons: Siguiente, Atrás, Finalizar, Cancelar.

Figura 2 Página que permite incluir los archivos y carpetas a exportar.

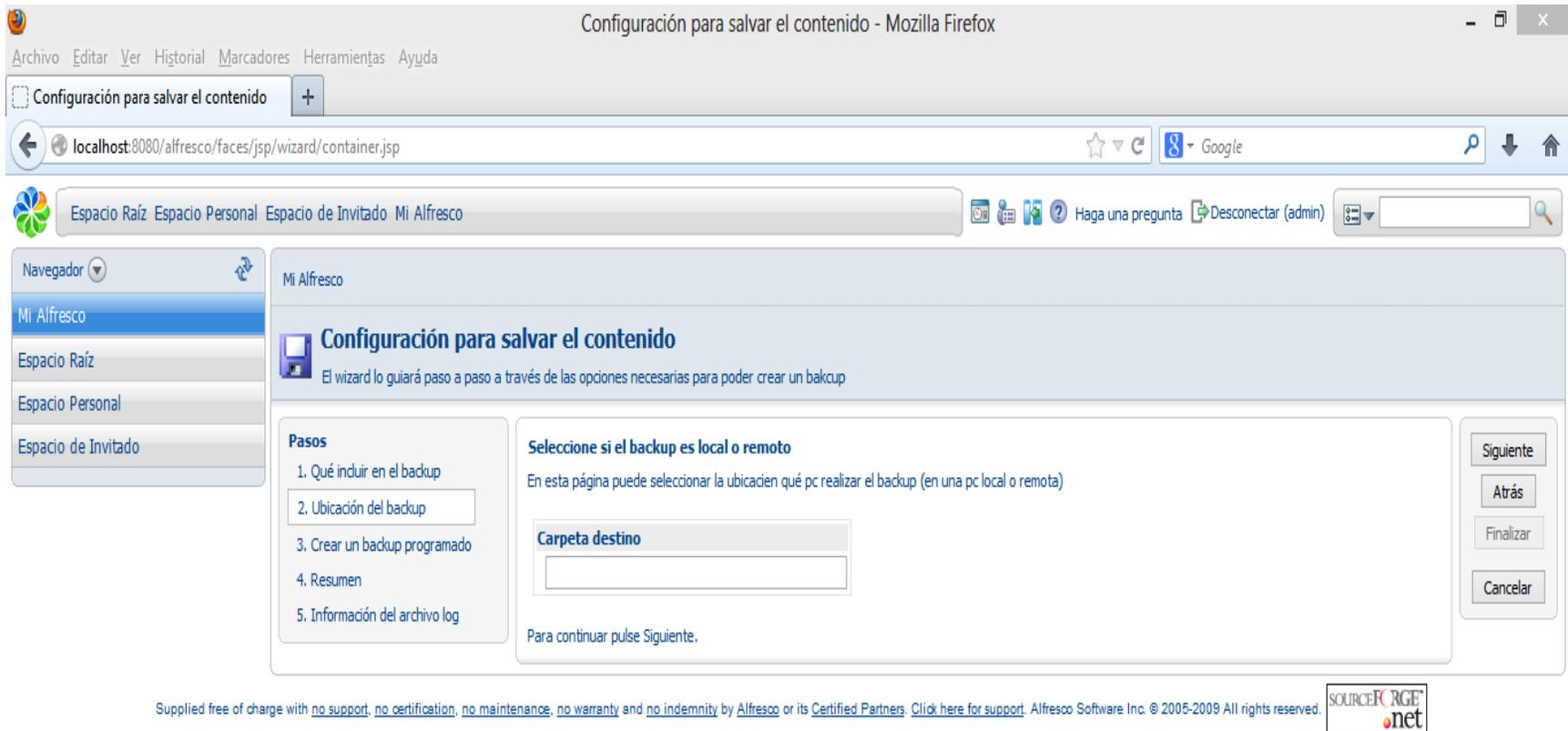


Figura 3 Página que permite escoger la ubicación de la copia de seguridad a realizar.

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

Configuración para salvar el contenido +

localhost:8080/alfresco/faces/jsp/wizard/container.jsp

Google

Espacio Raíz Espacio Personal Espacio de Invitado Mi Alfresco

Haga una pregunta Desconectar (admin)

Navegador

Mi Alfresco

Espacio Raíz

Espacio Personal

Espacio de Invitado

Mi Alfresco

Configuración para salvar el contenido

El wizard lo guiará paso a paso a través de las opciones necesarias para poder crear un backup

Pasos

1. Qué incluir en el backup
2. Ubicación del backup
3. Crear un backup programado
4. Resumen
5. Información del archivo log

Crear un backup programado

Ejecutar el backup ahora o programarlo

Realizar el backup ahora

Crear un backup programado

Fecha y Hora

Frecuencia: Cada semana

Dom	Lun	Mar	Mie	Jue	Vie	Sab
<input type="checkbox"/>						

Hora de comienzo: 08 : 47

Ejecutar el backup ahora o programarlo

Siguiete

Atrás

Finalizar

Cancelar

Supplied free of charge with no support, no certification, no maintenance, no warranty and no indemnity by Alfresco or its Certified Partners. [Click here for support](#). Alfresco Software Inc. © 2005-2009 All rights reserved.



Figura 4 Página que permite escoger la frecuencia de ejecución de las copias de seguridad programadas.

The screenshot shows a web browser window titled "Configuración para salvar el contenido - Mozilla Firefox". The address bar shows the URL "localhost:8080/alfresco/faces/jsp/wizard/container.jsp". The page content is for the "Configuración para salvar el contenido" wizard, which is guiding the user through the steps to create a backup. The current step is the "Resumen" (Summary) page, which displays the information entered by the user:

- Pasos:**
 1. Qué incluir en el backup
 2. Ubicación del backup
 3. Crear un backup programado
 4. Resumen
 5. Información del archivo log
- Resumen:**

A continuación se muestra la información que ha introducido:

 - Ubicación del backup:** c:/
 - Cuándo ejecutar el backup:** Se ejecutará los días Dom a las 08:47:00
 - Instalación de Alfresco:**
 - Carpeta de Alfresco: c:/excriba/
 - Carpeta de alf_data: \${excriba.repodir}/excriba-datas
 - Carpeta de los Índices de Lucene: \${excriba.repodir}/lucene-indexes
- Ejecutar el backup en background
 - El backups se ejecutará en background así que los resultados no aparecerán de forma inmediata

Presione el botón Siguiente para ejecutar el backup con las opciones seleccionadas

Navigation buttons: Siguiente, Atrás, Finalizar, Cancelar.

Footer: Supplied free of charge with no support, no certification, no maintenance, no warranty and no indemnity by Alfresco or its Certified Partners. Click here for support. Alfresco Software Inc. © 2005-2009 All rights reserved. SOURCEFORGE.net

Figura 5 Página que muestra el resumen de los datos introducidos para la exportación.

Backups realizados y programados - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

Backups realizados y programados

localhost:8080/alfresco/faces/jsp/dialog/container.jsp

Espacio Raíz Espacio Personal Espacio de Invitado Mi Alfresco

Haga una pregunta Desconectar (admin)

Navegador

- Mi Alfresco
- Espacio Raíz
- Espacio Personal
- Espacio de Invitado

Mi Alfresco

Backups realizados y programados

En esta página puede restaurar el backup seleccionado

▼ Backups programados Backups por páginas 10

Expresión cron	Día/Hora creado	Día/Hora modificado	Día/Hora ejecutará	Acciones
0 7 21 1 11 ? 2013	31/03/2013 21:08	31/03/2013 22:58	Se ejecutará el día 01/11/2013 a las 21:07	
0 8 21 ? * SUN,MON,TUE,WED	31/03/2013 21:08	01/04/2013 07:51	Dom,Lun,Mar,Jue 21:08	

Página 1 de 1

▼ Backups realizados Backups por páginas 10

Nombre del backup	Día/Hora ejecutado	Día/Hora terminado	Archivos/Sistema de Archivos	Local or Remota	Errores	Acciones
1_Backup_SR	17/03/2013 20:55:00	17/03/2013 21:26:51	Archivos	local	No	
2_Backup_SR	29/03/2013 06:51:10	29/03/2013 06:51:10	Archivos	local	No	
3_Backup_SR	30/03/2013 20:15:00		Archivos	local	Yes	

Página 1 de 1

Aceptar

Cancelar

Figura 6 Página que muestra los listados de copias de seguridad realizadas y programadas.

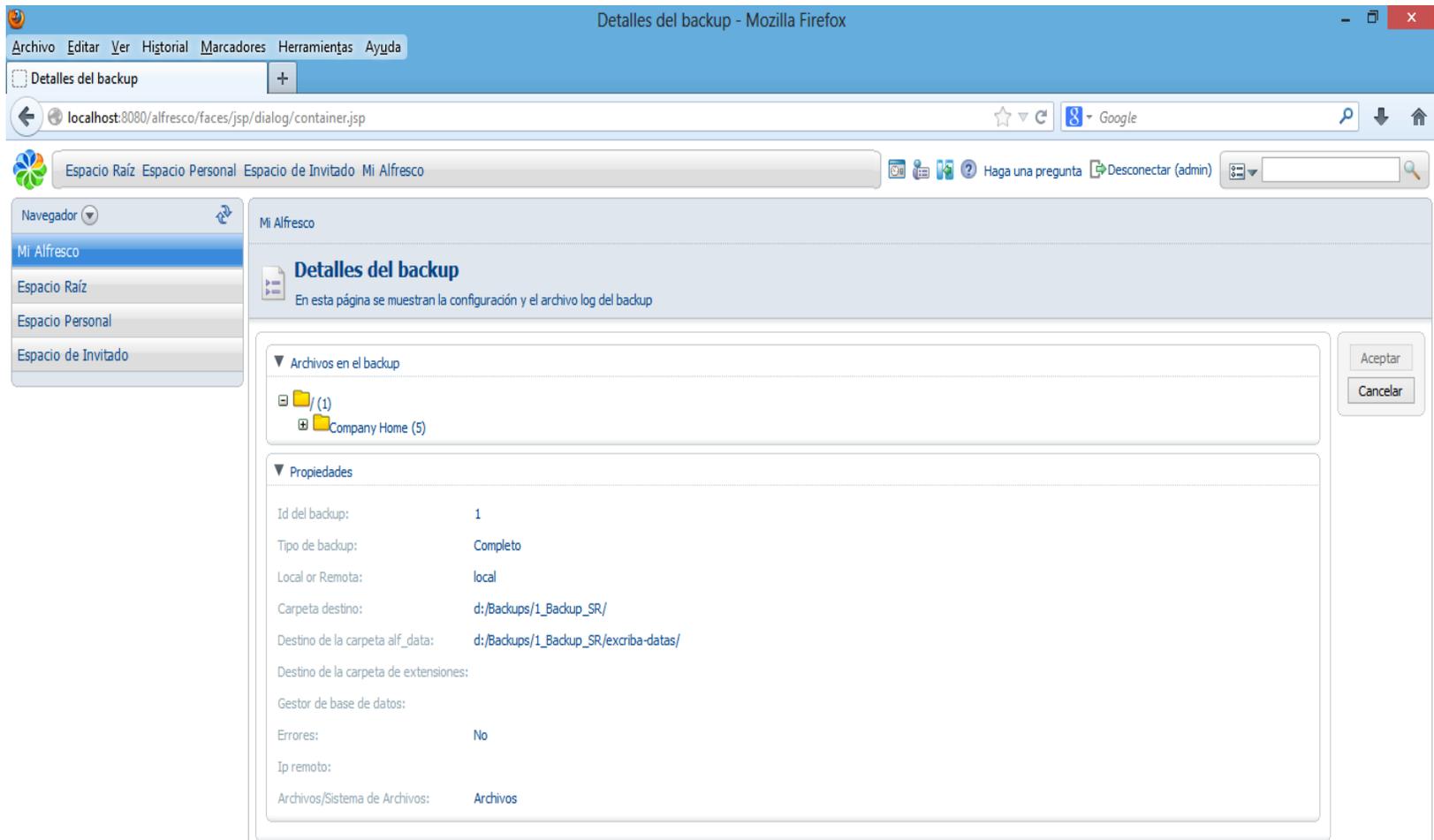


Figura 7 Página que muestra los detalles de la copia de seguridad realizada.

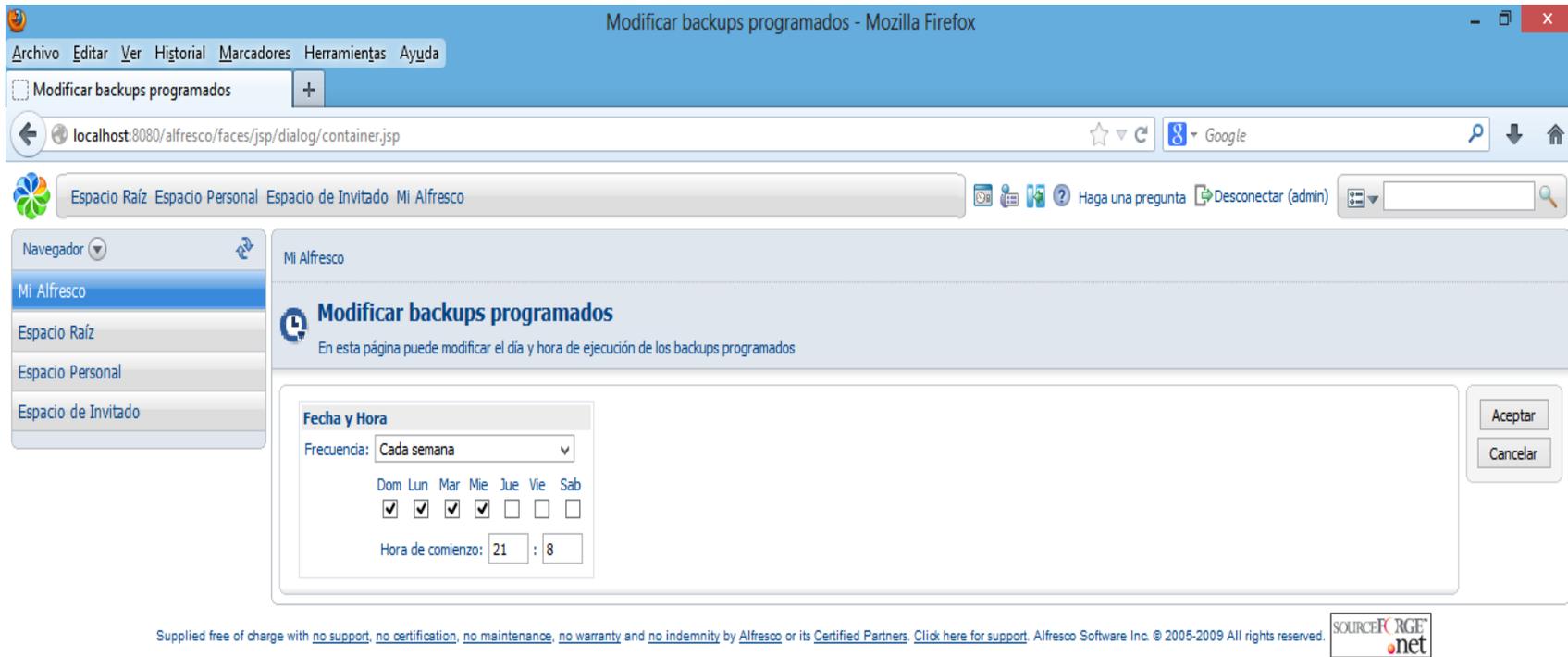


Figura 8 Página que permite modificar la frecuencia de ejecución de las copias de seguridad programadas.

Anexo3: Artefactos de Ingeniería de Software generados en la etapa de diseño de la solución propuesta

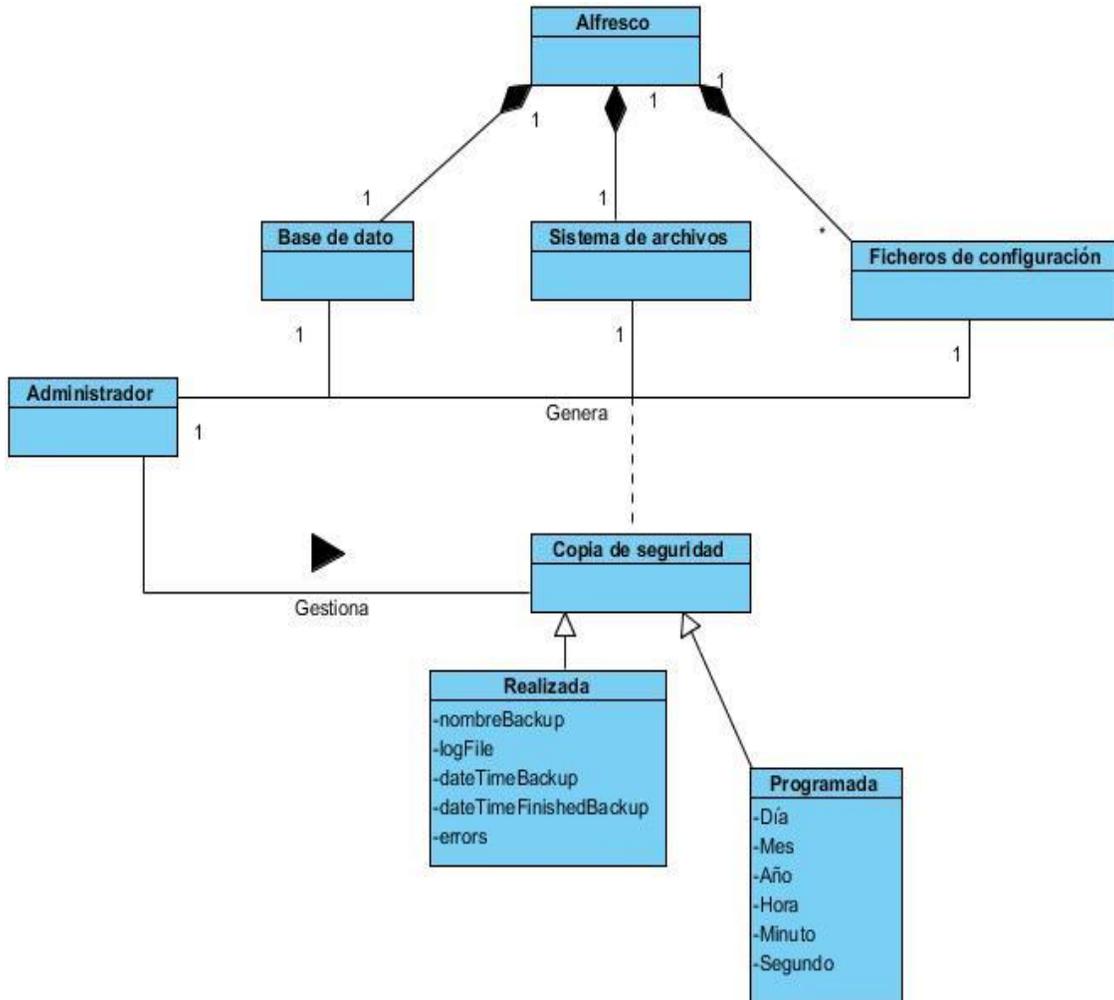


Figura 9 Modelo del dominio.

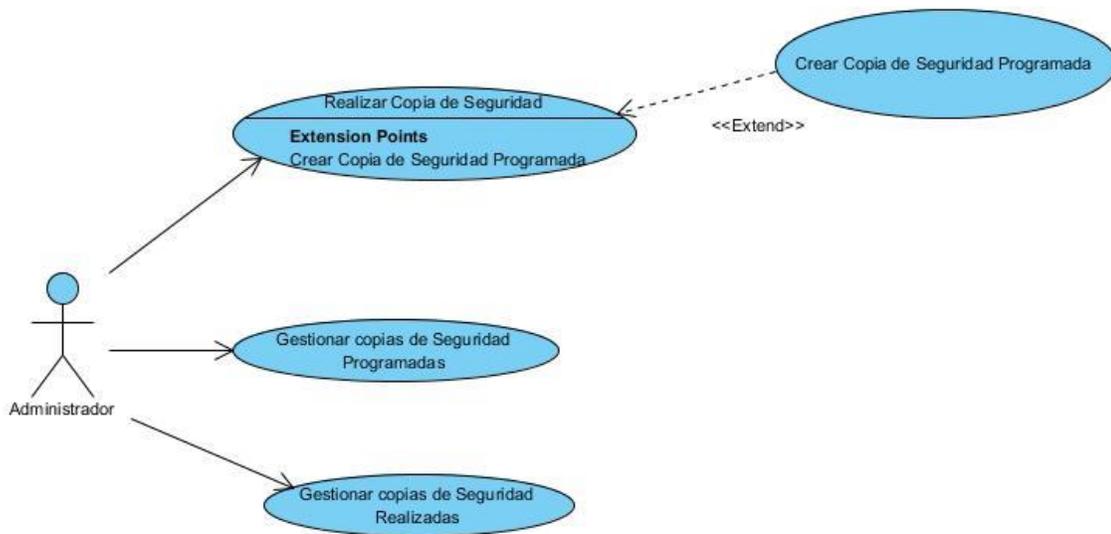


Figura 10 Diagrama de Casos de Usos del Sistema (CUS).

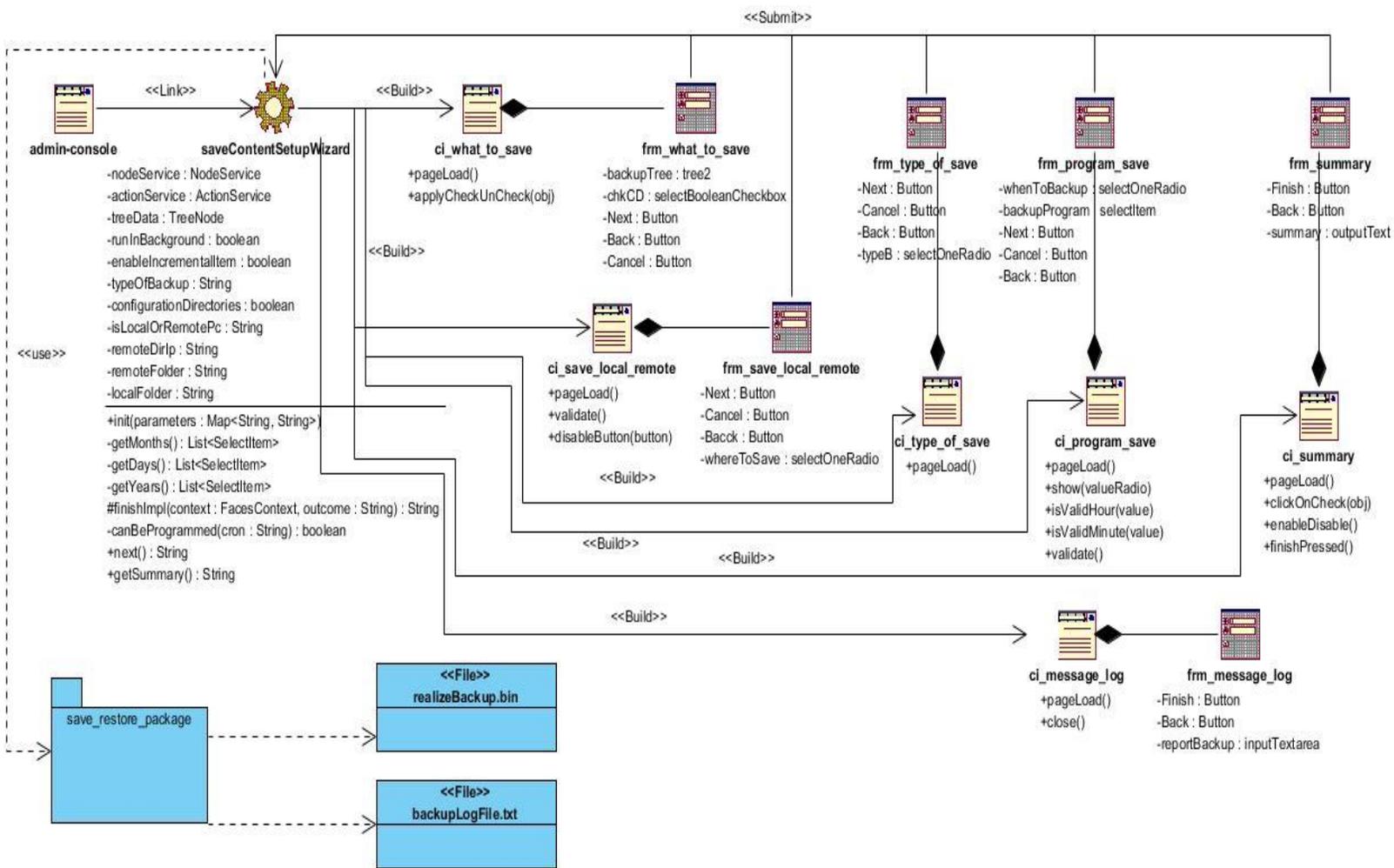


Figura 11 Diagrama de Clases correspondiente al CU "Realizar Copia de Seguridad".

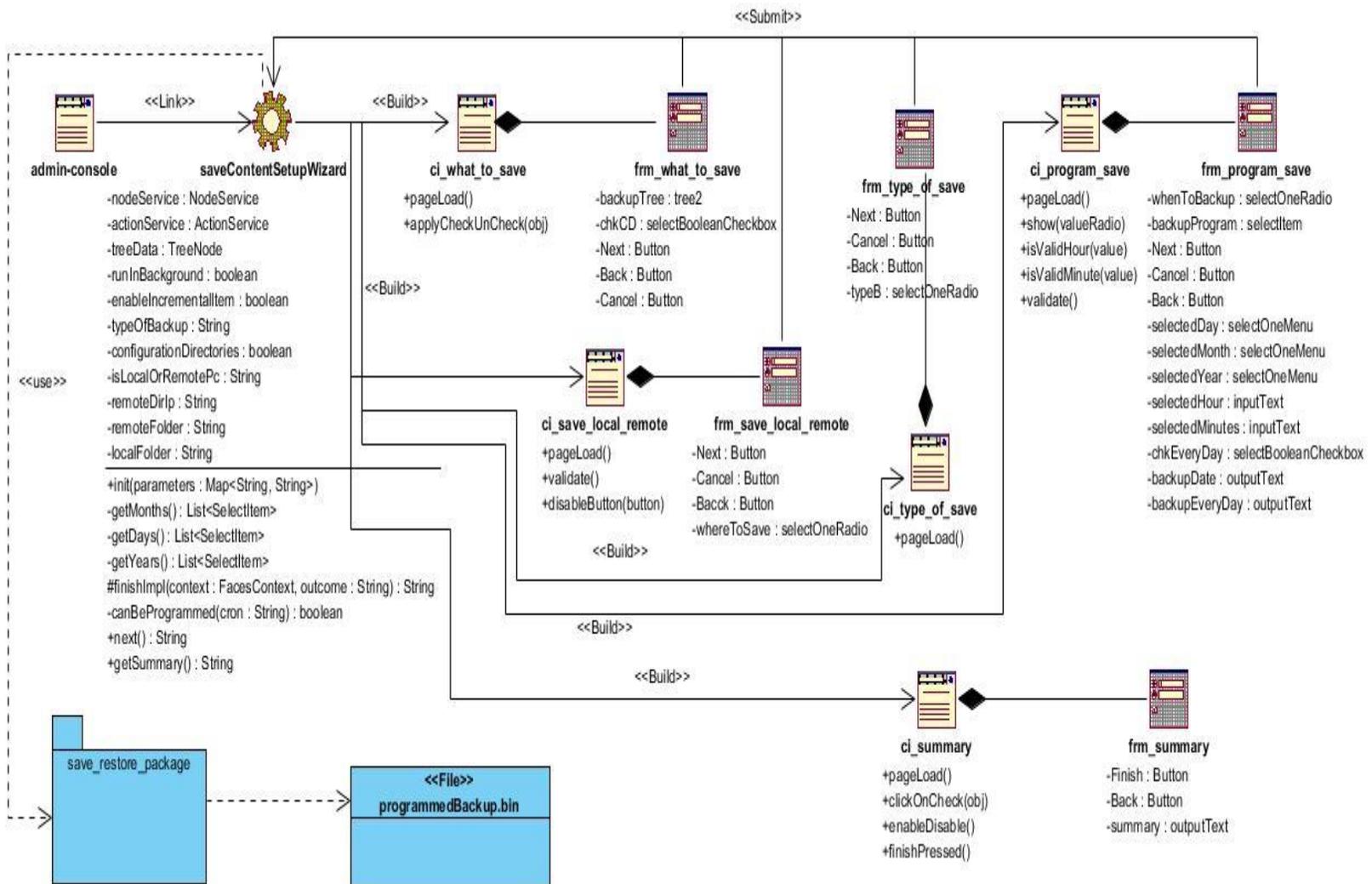


Figura 12 Diagrama de Clases correspondiente al CU "Crear Copia de Seguridad Programada".

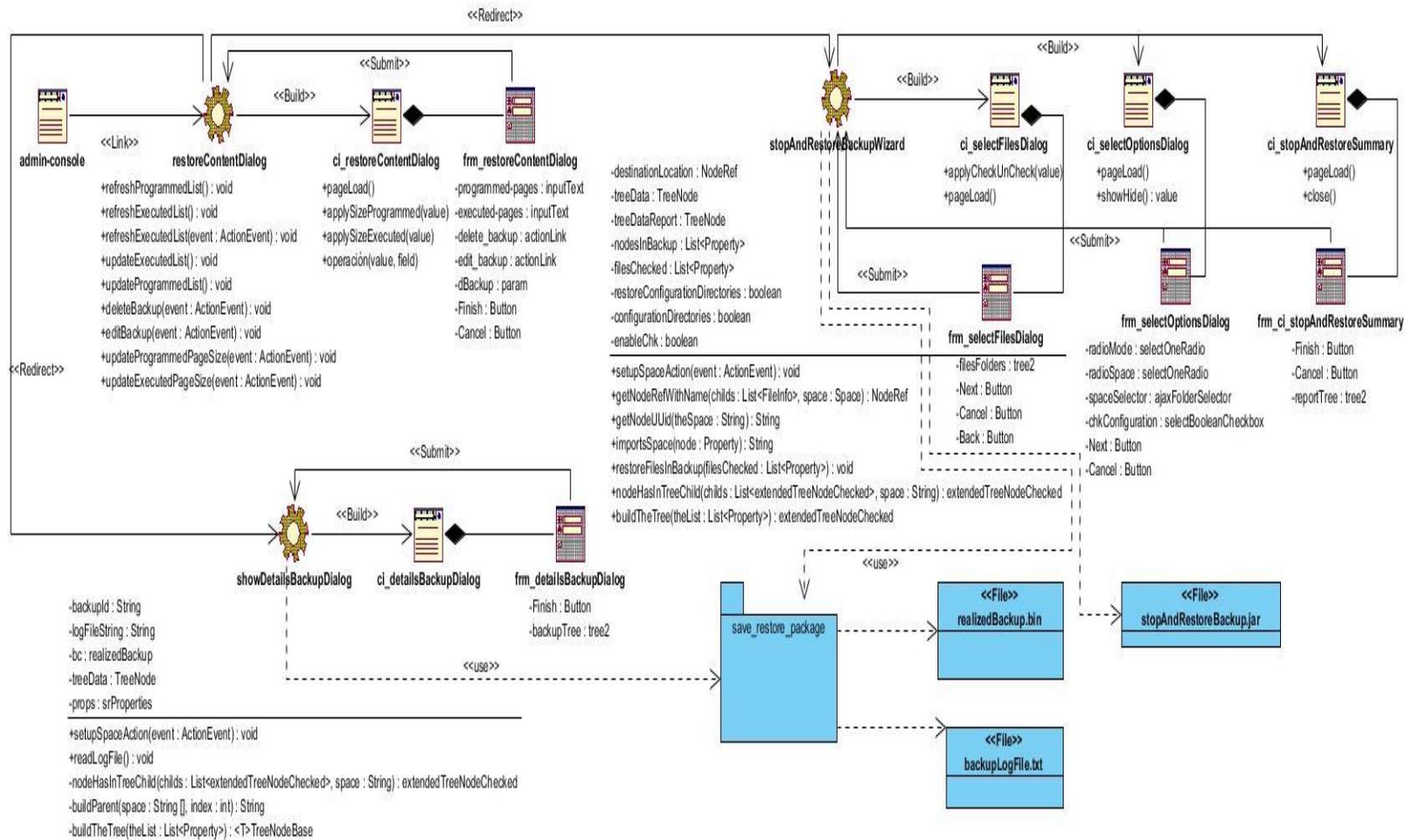


Figura 13 Diagrama de Clases correspondiente al CU “Gestionar Copias de Seguridad Realizadas”.

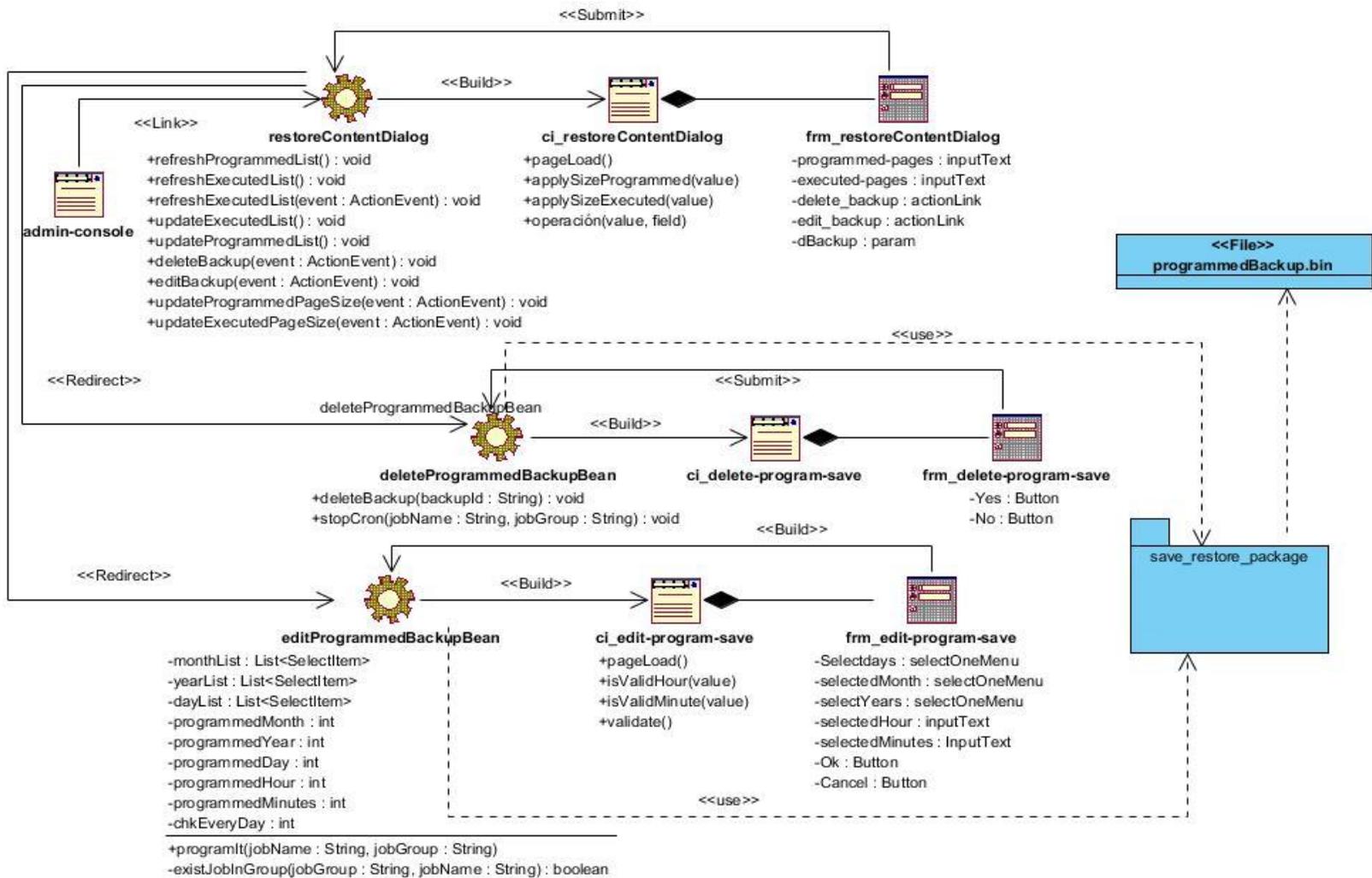


Figura 14 Diagrama de Clases correspondiente al CU "Gestionar Copias de Seguridad Programadas".

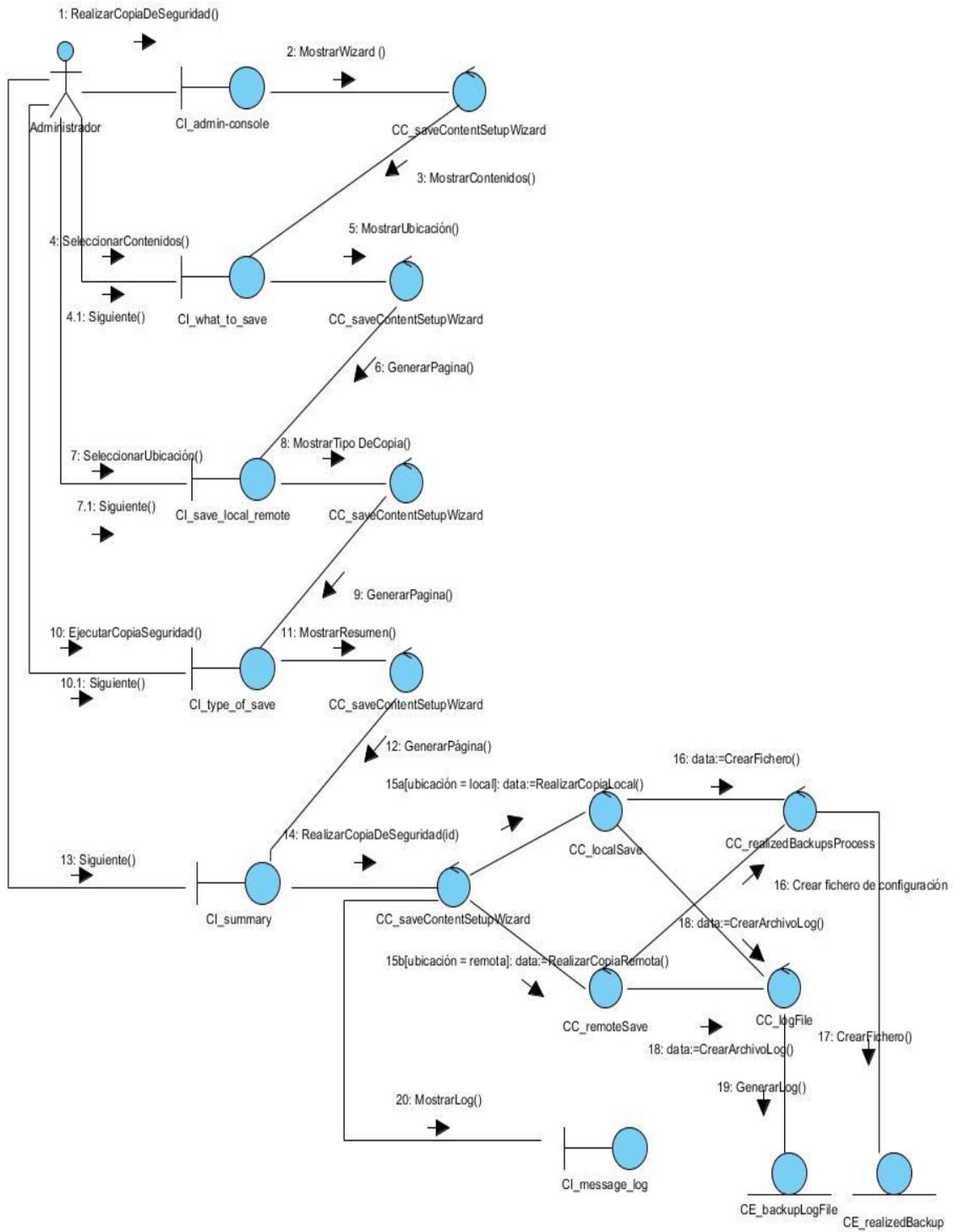


Figura 15 Diagrama de Colaboración correspondiente al CU “Realizar Copia de Seguridad”.

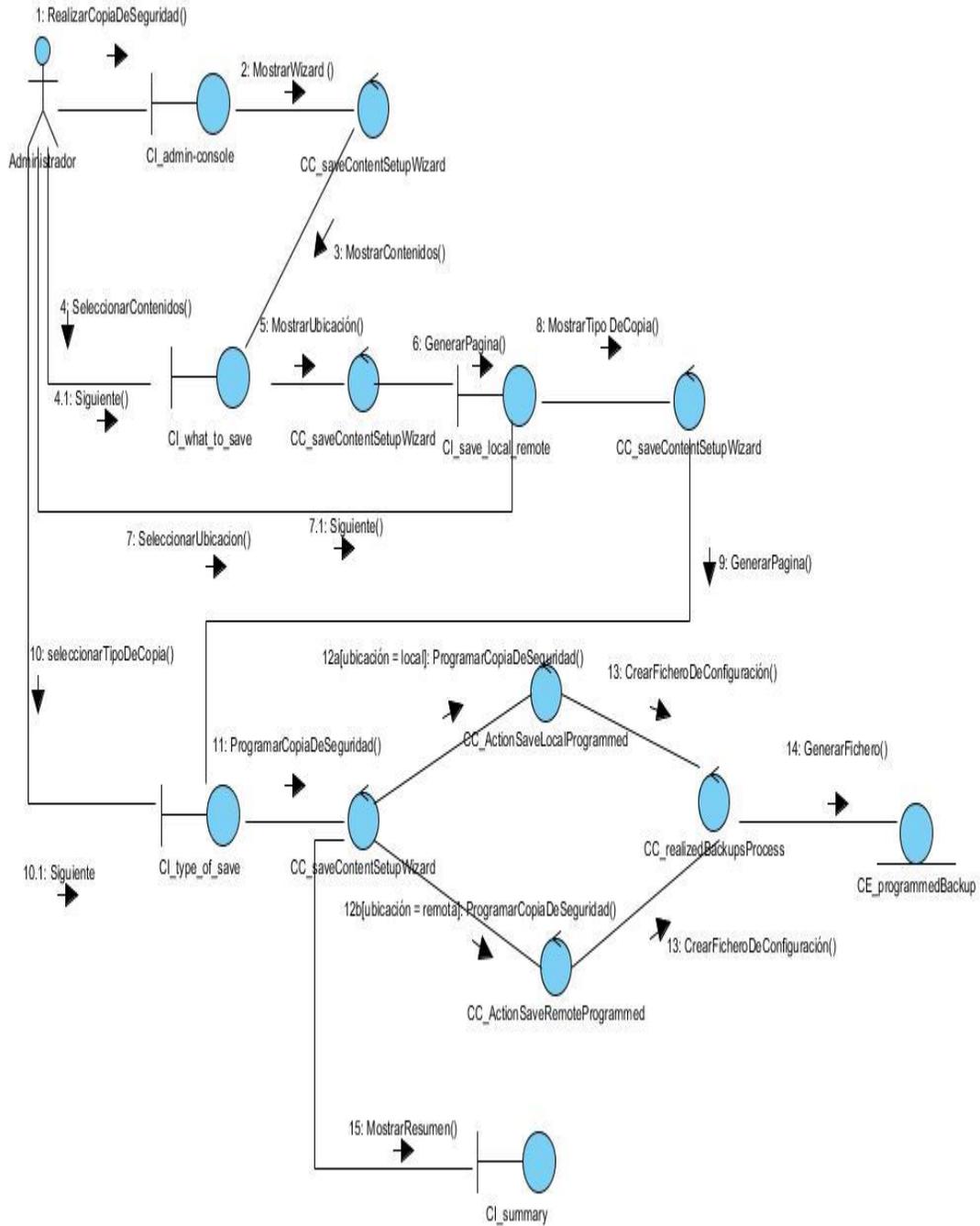


Figura 16 Diagrama de Colaboración correspondiente al CU “Crear Copia de Seguridad Programada”.

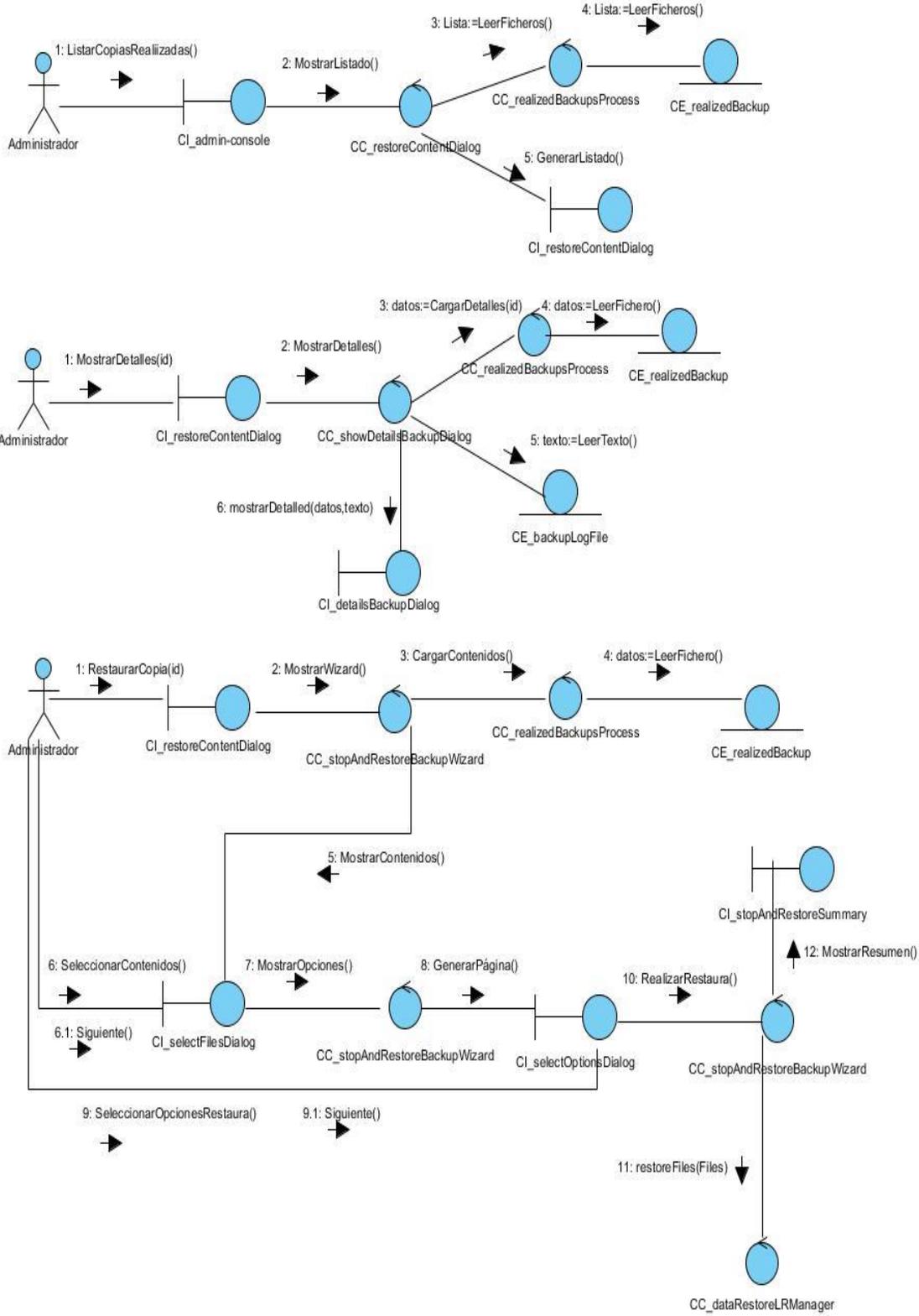


Figura 17 Diagrama de Colaboración correspondiente al CU "Gestionar Copias de Seguridad Realizadas".

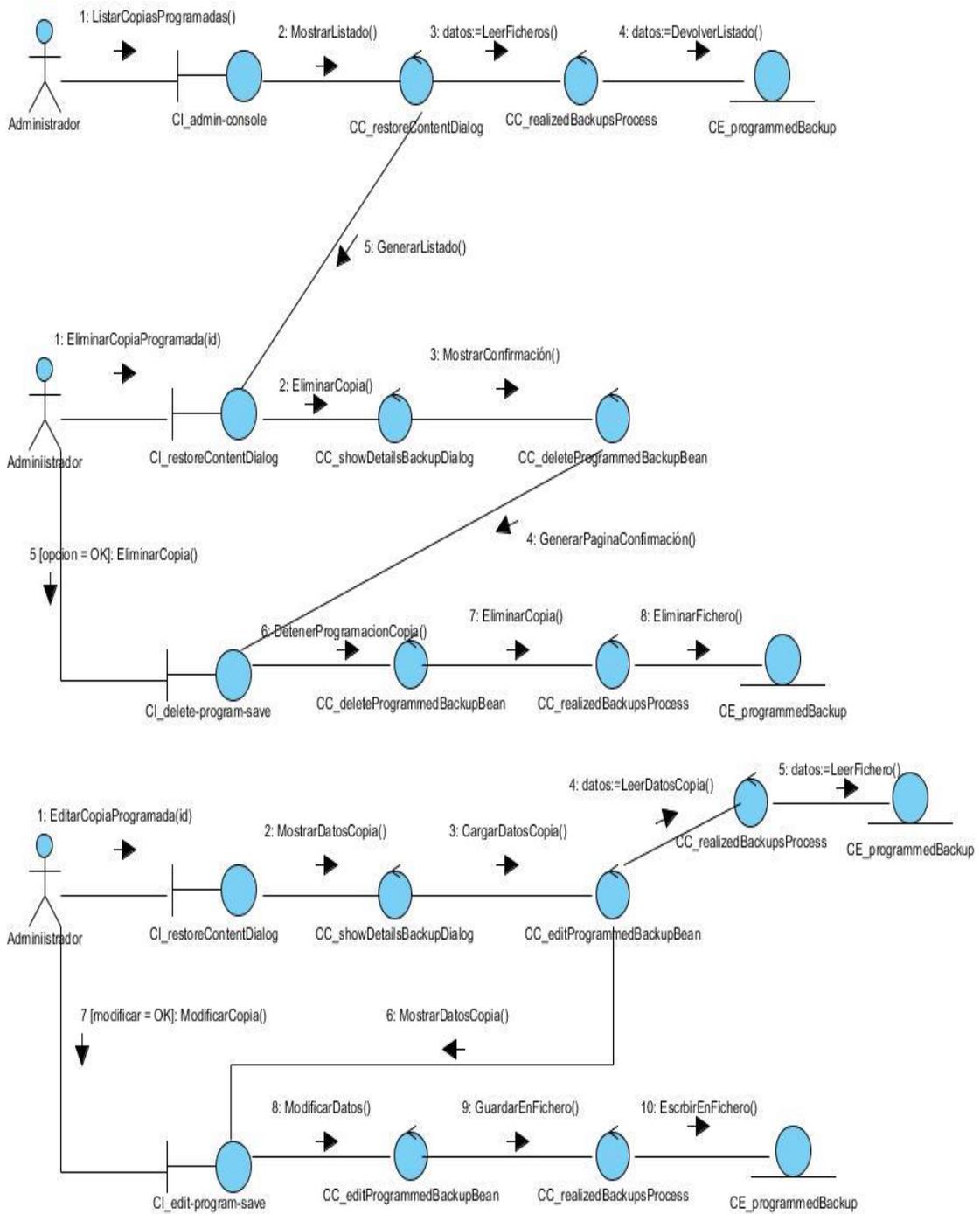


Figura 18 Diagrama de Colaboración correspondiente al CU "Gestionar Copias de Seguridad Programadas".

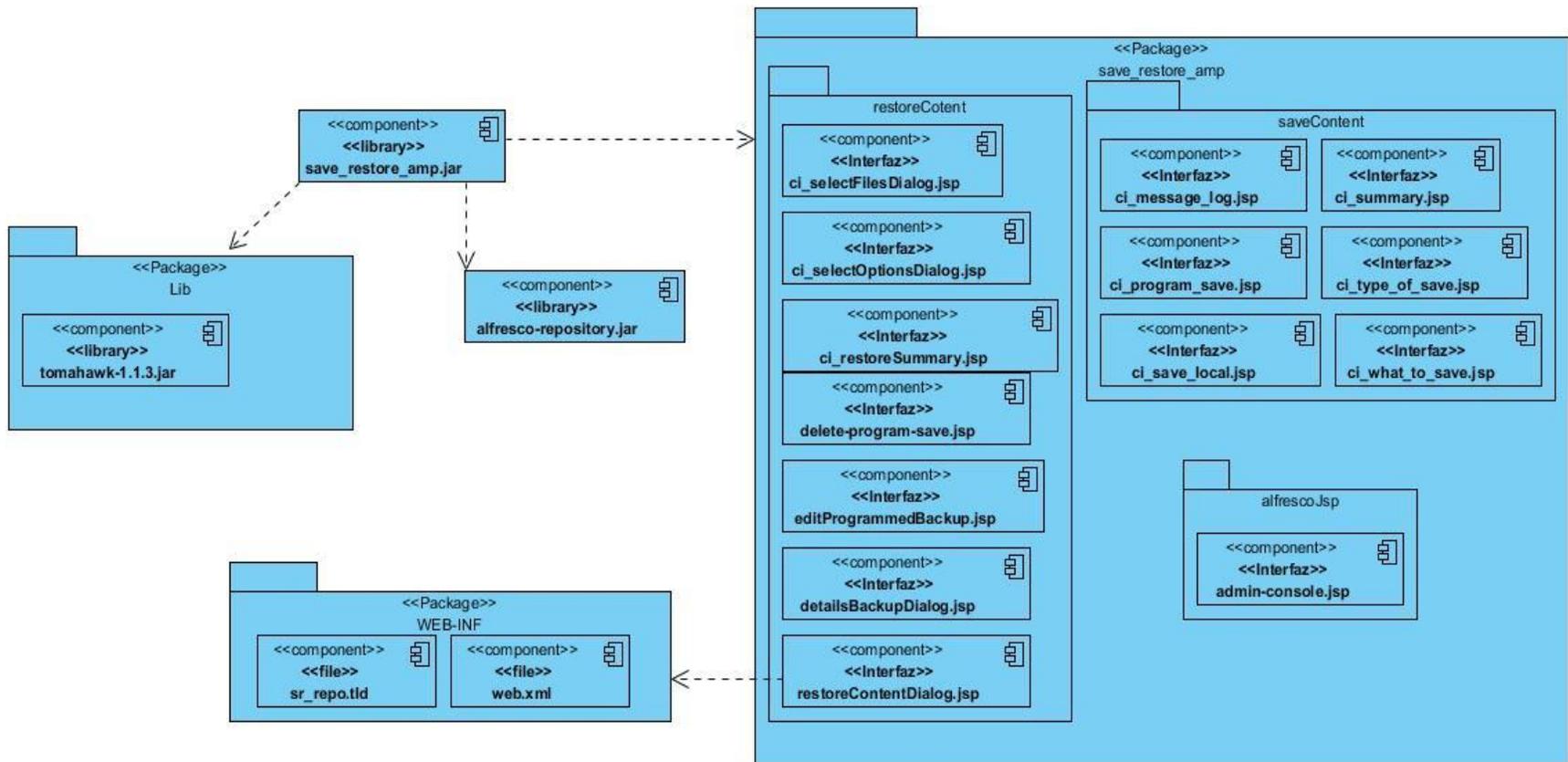


Figura 19 Diagrama de Componentes de la solución propuesta para la Gestión de Copias de Seguridad.

