

Universidad de las Ciencias Informáticas

Facultad 1



Módulo para importar y exportar agrupaciones documentales y sus descripciones.

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

Autores: Danilo Alberto Caballero Rodríguez
Rolando Rodríguez Junco

Tutor: Ing. Yenisel Valdés Hernández

Co-Tutor: Ing. Reinier Elejalde Chacón

Ciudad de la Habana, 2011

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la UCI los derechos patrimoniales de la misma, con carácter exclusivo.

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

Danilo Alberto Caballero Rodríguez

Rolando Rodríguez Junco

Cargo del tutor
Ing. Yenisel Valdés Hernández

Cargo del co-tutor
Ing. Reinier Elejalde Chacón

Resumen

El proceso de conservación de la información en nuestros días adquiere vital importancia cuando todo tipo de negocio requiere de esta para trazarse futuras metas y expectativas. Debido al rápido crecimiento del volumen de información en las instituciones es necesario el uso de estándares y normas que regulen la forma en que se presenta, gestiona, almacena y comparte.

Además con el desarrollo de las tecnologías de la información y el rápido auge de las redes de comunicación, se necesitan sistemas especializados en el manejo de la información. Centrándose en el análisis del comportamiento en los sistemas de gestión documental y archivística los cuales se encargan de facilitar la administración de la información que se almacena en ellos.

Como consecuencia de la variedad de criterios entre diferentes regiones y países en cuanto a la terminología que usan para la descripción de sus documentos, la norma ISAD(G), unifica parte de los criterios existentes logrando homogeneidad en la información. Así mismo el estándar de descripción de archivos codificados EAD se propone como un medio de almacenamiento de las descripciones de los documentos pertenecientes a las instituciones. La norma ISAD(G) y el estándar EAD están estrechamente relacionados debido a que el estándar EAD es el medio mediante el cual la unificación de criterios lograda con la norma ISAD(G) se puede consultar, extraer, presentar e intercambiar entre los sistemas de gestión documental que implementen dicho estándar.

En el trabajo que se presenta a continuación se trata de mostrar los elementos necesarios para hacer posible la implementación de un software que sea capaz de intercambiar agrupaciones documentales y sus descripciones para el SGDA eXcriba.

Índice General

Índice de figuras	VI
Introducción	1
1. Fundamentación teórica	5
1.1. Conceptos Generales	5
1.1.1. Gestión Documental	5
1.1.2. Archivística	7
1.1.3. Archivo	7
1.1.4. Clasificación de los Archivos	8
1.1.5. La Descripción de Documentos de Archivo	9
1.1.6. Niveles de Jerarquía	9
1.1.7. Los instrumentos de descripción	10
1.2. Los estándares y las normas internacionales	10
1.2.1. Norma ISAD (G)	11
1.2.2. Estándar EAD	13
1.3. Estado del Arte	16
1.4. Herramientas, tecnologías y metodologías utilizadas	18
1.4.1. Aplicación de escritorio	18
1.4.2. Lenguaje de programación	18

1.4.3.	Entorno de desarrollo integrado	20
1.4.4.	Metodología de desarrollo de software	21
1.4.5.	Lenguaje para modelar los artefactos	24
1.4.6.	Herramientas CASE	25
2.	Características del sistema	29
2.1.	Modelo de dominio	29
2.2.	Propuesta del sistema	31
2.3.	Especificación de requisitos	32
2.3.1.	Requisitos funcionales	32
2.3.2.	Requisitos no funcionales	33
2.4.	Descripción de los actores del sistema	34
2.5.	Diagrama de casos de uso del sistema	35
2.5.1.	Definición de los Casos de Uso	35
3.	Diseño del subsistema	39
3.1.	Diseño de la aplicación	39
3.1.1.	Diagramas de interacción	40
3.1.2.	Diagrama de clases del diseño	42
4.	Implementación y prueba	53
4.1.	Implementación	53
4.1.1.	Diagrama de despliegue	53
4.1.2.	Diagrama de componentes	54
4.1.3.	Estándar de codificación	55
4.1.4.	Pruebas del sistema	58
	Conclusiones	63

Recomendaciones	64
Referencias bibliográficas	67
Bibliografía	69

Índice de figuras

2.1. Modelo de dominio.	30
2.2. Diagrama de Casos de Uso del Sistema.	35
3.1. Diagrama de Colaboración CUS 1: Autenticar usuario.	40
3.2. Diagrama de Colaboración CUS 2: Cerrar sesión.	41
3.3. Diagrama de Colaboración CUS 3: Importar contenidos desde un estándar EAD.	41
3.4. Diagrama de Colaboración CUS 4: Exportar contenidos en el estándar EAD.	41
3.5. Diagrama de clases de diseño del paquete cu.uci.geitel.gda.excriba.enarde.visual.	42
3.6. Diagrama de clases de diseño del paquete cu.uci.geitel.gda.excriba.enarde.codigo.	43
4.1. Diagrama de Despliegue.	54
4.2. Diagrama de Componentes.	55

Introducción

Gestión documental es *“el conjunto de normas, técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización, permitir la recuperación de información desde ellos, determinar el tiempo que los documentos deben guardarse, eliminar los que ya no se utilizan y asegurar la custodia indefinida de los documentos más valiosos.”* [1]

En la Universidad de las Ciencias Informáticas (UCI), en la cual se produce software y se brindan servicios informáticos, el departamento de Gestión Documental y Archivística de la Facultad 1 desarrolla el software eXcriba ¹.

Gran parte de la documentación que se almacena en el eXcriba se organiza dentro del cuadro de clasificación, el cual está sujeto a una estructura jerárquica que define la institución en que se encuentra. Luego del despliegue del sistema en varias instituciones a nivel nacional, se comprendió la necesidad de transferir sus agrupaciones documentales y descripciones hacia otras instituciones encargadas de preservar su patrimonio documental.

Este proceso de transferencia de información se realiza manualmente y se hace bastante difícil debido a que se debe conocer el lugar específico donde se coloca cada documento electrónico en el cuadro de clasificación cuando este sea movido fuera de la institución para que no pierda su estructura original y se deben transferir todos los datos asociados a las descripciones de cada agrupación documental. La pérdida de tiempo que provoca realizar este proceso de forma manual es muy grande para los encargados del mismo, además no se puede garantizar la integridad de la información por ser indetectable el mal manejo por error humano.

Por todo lo antes planteado se llega a la conclusión de que el sistema eXcriba no posee una vía para importar y exportar agrupaciones documentales y sus descripciones entre instituciones, para su procesamiento

¹Gestor de Documentos Administrativos.

o almacenamiento en un lugar que perdure en la historia y se le puedan realizar futuras consultas.

A partir de la situación problemática anterior se plantea la siguiente interrogante:

¿Cómo importar y/o exportar agrupaciones documentales y sus descripciones entre el sistema gestor de documentos administrativos eXcriba y cualquier otro sistema?

Para enfocar esta investigación se define como **objeto de estudio** los estándares para la Descripción Archivística que existen actualmente en el mundo como medio para representar datos, delimitando el **campo de acción** en las Normas para importar y exportar agrupaciones documentales y sus descripciones.

Se trazó como **objetivo general**:

Desarrollar un módulo para la importación y exportación de agrupaciones documentales y sus descripciones para el software eXcriba.

Con el propósito de darle cumplimiento al mismo se definen los siguientes objetivos específicos:

- Realizar un estudio bibliográfico sobre la gestión documental y las normas internacionales.
- Efectuar la búsqueda de aplicaciones que se dediquen a importar y exportar agrupaciones documentales y sus descripciones.
- Realizar el diseño del módulo a implementar.
- Implementar un módulo que facilite importar y/o exportar agrupaciones documentales y sus descripciones.
- Validar la calidad de la aplicación mediante pruebas de caja negra.

El trabajo de diploma queda sustentado en la siguiente **idea a defender**: el desarrollo de un módulo para la importación y exportación de agrupaciones documentales y sus descripciones posibilitará su intercambio entre sistemas de forma rápida y eficiente.

Para el cumplimiento de dichos objetivos se han definido las siguientes tareas de la investigación:

- Revisión bibliográfica sobre la gestión documental y los tipos de archivos.

- Análisis de la Norma de Descripción Archivística (ISAD(G)²).
- Análisis de estándares utilizados para la representación de datos (EAD³, EAC⁴).
- Estudio de la Definición de Tipo de Documento (DTD ⁵) para EAD.
- Análisis de la relación entre las normas de descripción archivística y los estándares de representación de metadatos.
- Estudio de las herramientas y tecnologías que serán empleadas en el desarrollo de la solución.
- Determinación de los requisitos funcionales y no funcionales del sistema.
- Elaboración del diseño del módulo propuesto.
- Implementación de un módulo que permita el intercambio de documentos para el Sistema de Gestión Documental eXcriba.
- Ejecución de las prueba de caja negra para validar la calidad de la aplicación.

Los métodos de investigación a utilizar son:

Analítico-Sintético: para la realización del presente trabajo de diploma se extraerán del objeto de estudio conceptos que serán analizados por separado confrontando los criterios de diferentes autores con el objetivo de comprender cada uno de forma individual. Además se analizarán los procesos relacionados con el uso de los estándares de descripción, para lograr la asimilación de los elementos más importantes relacionados con estos. También se hará un estudio de las diferentes herramientas y tecnologías para conocer las diferencia entre estas y lograr seleccionar las más adecuadas. Una vez terminado el análisis se brindará una propuesta de solución del problema.

²Norma Internacional General de Descripción Archivística.

³Estándar de Descripción Archivística Codificada (Encoded Archival Description).

⁴Estándar Contexto Archivístico Codificado (Encoded Archival Context).

⁵Descripción de estructura y sintaxis de un documento XML o SGML.

Modelado: con la intención de ayudar a comprender las teorías relacionadas con el estándar EAD se modeló de forma simplificada su estructura, manteniendo una correspondencia con la sintaxis original del estándar. Descomponiéndolo en un modelo operativo mucho más fácil de estudiar que el fenómeno real.

El presente trabajo tendrá cuatro capítulos estructurados de la siguiente forma:

Capítulo 1 “Fundamentación teórica” En este capítulo se abordarán los conceptos teóricos relevantes que serán necesarios para comprender la investigación. Se realizará un análisis de cuales serán las herramientas, tecnologías y lenguajes a utilizar para implementar el módulo. Se pretende ayudar a comprender mediante la investigación, como será posible la implementación de un módulo que permita exportar e importar agrupaciones documentales y sus descripciones entre sistemas.

Capítulo 2 “Características del subsistema” Se hará una descripción de la propuesta de desarrollo y se expondrán cada uno de los procesos del negocio así como los sujetos que intervienen en los mismos. También serán expuestos los requisitos para desarrollar el módulo y cada uno de los casos de uso del mismo.

Capítulo 3 “Diseño del subsistema” Se especificarán los detalles de la construcción de la solución; se modelarán los diagramas de clases y quedará representada la arquitectura definida, así como los diferentes patrones a utilizar.

Capítulo 4 “Implementación y prueba” Se representarán gráficamente los diagramas de despliegue y de componentes. Una vez terminada la implementación del módulo se documentará el proceso de las pruebas de calidad.

Capítulo 1

Fundamentación teórica

En el presente capítulo se pretende hacer una investigación exhaustiva que constituya la base para la elaboración de un módulo para exportar e importar agrupaciones documentales y sus descripciones en el sistema **eXcriba**. Con el apoyo en conceptos teóricos acerca del tema se podrá entender la investigación y su propósito. Se analizarán las herramientas, tecnologías y lenguajes a utilizar para implementar el módulo.

1.1. Conceptos Generales

1.1.1. Gestión Documental

La Gestión Documental se define como *"...una parte del sistema de información de la empresa desarrollado con el propósito de almacenar y recuperar documentos, que debe estar diseñado para coordinar y controlar todas aquellas funciones y actividades específicas que afectan a la creación, recepción, almacenamiento, acceso y preservación de los documentos, salvaguardando sus características estructurales, y contextuales, y garantizando su autenticidad y veracidad"*[2].

También se entiende por gestión documental el conjunto de normas utilizadas para administrar el flujo de documentos almacenados en una organización. Debe permitir la recuperación de información desde los documentos, determinar el tiempo de almacenamiento de estos, eliminar los que ya no sean necesarios y asegurar la conservación de los documentos más valiosos. Define tareas y procedimientos orientados a lograr

eficiencia en la creación, el mantenimiento, el uso y la disposición de los documentos de una organización durante todo su ciclo de vida y de hacer que la información que contienen esté disponible en apoyo del quehacer de esa organización y de los usuarios externos a ella que lo requieran.

La gestión de documentos “*comprende: planificación, control, dirección, organización, promoción y otras actividades de gestión relativas a los documentos antes de llegar a los archivos históricos, incluyendo el manejo de la correspondencia, formularios, microformas, técnicas de la automatización de datos, etc*” [3].

Con los adelantos de la tecnología y con ellas los documentos electrónicos, la sociedad se ha visto forzada a tratar con un soporte documental “desmaterializado”. La necesidad de seguir teniendo el documento como soporte de todas las transacciones que realicen las entidades han planteado la exigencia de encontrar un tratamiento al documento electrónico y al mismo tiempo aparece la necesidad de crear un sistema de gestión electrónica de documentos.

Beneficios de la Gestión Documental de forma electrónica

La Gestión Documental de forma electrónica brinda beneficios desde cualquier punto de vista de la organización. El tiempo es vital, y mientras que se invierte tiempo buscando, copiando o enviando papel: archivos, faxes, copias, etc., la gestión documental de forma electrónica se traduce en una mayor productividad.

Los sistemas de gestión documental hacen posible que los documentos en papel no sean necesarios y que sólo se trabaje con documentos digitales. Hay que tener en cuenta además que el papel cuesta dinero y espacio de almacenamiento, con “*un CD-ROM puede almacenar 120.000 páginas de listados o 15.000 páginas escaneadas*” [4]. La reducción en el coste de insumos, la disponibilidad de la información y mejoras en los tiempos de respuesta, resumen el por qué se hace necesario el empleo de la Gestión Documental y eliminan:

- Pérdida de tiempo en archiveros y carpetas.
- Aumento de costos por constante duplicación.

- Susceptibilidad a traspapeleo ¹, deterioro o pérdida del documento.
- No disponibilidad del documento.
- No disponibilidad de la multclasificación (se ordenan por período, orden alfabético o tipo).

1.1.2. Archivística

Son muchas las definiciones que se han emitido de archivística, todas coincidentes en que su objetivo son los archivos. Entre los conceptos más relevantes está el emitido en el diccionario de terminología Archivística que la define como la *“disciplina que trata de los aspectos teóricos y prácticos (tipología, organización, funcionamiento, planificación, etc.) de los archivos y el tratamiento archivístico de sus fondos documentales”* [5].

Concluyendo se define archivística como la ciencia que estudia la naturaleza de los archivos, los principios de su conservación y organización y los medios para su utilización. Se ocupará de la creación, organización, conservación y tratamiento de los documentos que en los archivos se almacenan.

1.1.3. Archivo

Según el Diccionario de Archivística, un archivo se define como: *“Conjunto de documentos, sea cuales sean su fecha, su forma y su soporte material, producidos o recibidos por toda persona física o moral, y por todo servicio u organismo público o privado, en el ejercicio de su actividad, y son, ya conservados por sus creadores o por sus sucesores para sus propias necesidades, ya transmitidos a la institución de archivos competente en razón de su valor archivístico”*[6].

Es considerado como archivo además aquella *“Institución responsable de la acogida, tratamiento, inventario, conservación y servicio de los documentos”*[6] es decir el depósito donde se guardan organizada y ordenadamente los documentos para conservarlos con el doble fin de garantizar los servicios de los administrados o de servir de fuentes para la investigación.

¹Extraviar un papel o un documento poniéndolo en un lugar que no le corresponde.

Muy ligado al archivo está el documento de archivo que es todo registro de información independiente de su lenguaje o soporte (forma oral o escrita, textual o gráfica, manuscrita o impresa, en lenguaje natural o codificado, en cualquier soporte documental así como en cualquier otra expresión gráfica, de sonido, de imagen o electrónica) que abarca todo lo que es transmitido a través del conocimiento humano. Es el testimonio material de una actividad humana realizada en el ejercicio de sus funciones por instituciones o personas físicas.

1.1.4. Clasificación de los Archivos

Según Antonia Heredia en su libro "Archivística General Teoría y Práctica", se deben tener en cuenta dos clasificaciones de archivos fundamentales: los archivos administrativos y los archivos históricos. Se realiza esta clasificación a partir de la incidencia de las problemáticas pertenecientes a cada institución en específico y el especial tratamiento de la documentación y la información en unos y otros. También puede encontrarse en algunos casos un archivo entre los dos anteriores conocido como Archivo intermedio, viendo en este caso que el archivo de gestión y el archivo intermedio forman parte del archivo administrativo.

Los Archivos Administrativos

Los Archivos Administrativos son "*archivo vigente, activo, que guarda con carácter temporal los documentos que se encuentran al servicio exclusivo de las instituciones a la cual pertenece*" [7]. Entre sus funcionalidades se encuentra la de salvaguardar los derechos de la institución que lo crea, facilitando su gestión. Los archivos administrativos son los que inician o reciben los documentos en los primeros momentos de estos y deben permanecer cerca de los usuarios del archivo. Al archivo administrativo también se le llama archivo vivo.

Entre sus funcionalidades se encuentran la organización y puesta en servicio de la documentación administrativa, durante su periodo de máxima utilidad para la gestión administrativa. Aseguran la transferencia periódica de los documentos que ya no son de valor administrativo, facilitando y agilizando la recuperación de los documentos.

Los Archivos Históricos

Los Archivos Históricos constituyen el conjunto de escritos que han perdido interés administrativo para las oficinas que los han producido, han adquirido madurez archivística y han sido seleccionados para la conservación permanente.

Un archivo histórico es el encargado de custodiar y organizar el patrimonio documental, cuya función se resume en: recoger, conservar y servir. No se limitan a la materialidad de la recogida y al almacenamiento, sino que exigen un control y canalización de la producción documental, una conservación no sólo física sino que mira a la perdurabilidad y el servicio que conllevará difusión, orientación, ofrecimiento de datos concretos, facilidad de acceso y consulta.

1.1.5. La Descripción de Documentos de Archivo

La Descripción de Documentos de Archivo es la *“representación precisa de una unidad de descripción y, en su caso, de las partes que la componen, obtenida mediante la recopilación, cotejo, análisis y organización de toda aquella información que sirva para identificar, gestionar y localizar la documentación y explicar su contenido y el contexto de su producción”*[8].

Esta descripción se puede clasificar también como *“el medio utilizado por el archivero para obtener la información contenida en los documentos y ofrecerla a los interesados en ella”*[9]. Con la descripción de documentos se buscan dos objetivos: brindar información al usuario en la consulta, como vínculo o intermediario entre los usuarios de los archivos y la información contenida en sus acervos y facilitar el control los mismos.

1.1.6. Niveles de Jerarquía

Para que el Archivo pueda realizar su cometido es imprescindible que la documentación que resguarda pueda ser localizada con facilidad y ello no es posible si no se clasifica y ordena. La clasificación consiste en separar los elementos en conjuntos, estableciéndolos como grupos diferentes (fondo, secciones y series),

respetando la procedencia y bajo el esquema del cuadro de clasificación. Ordenar significa poner los elementos en el lugar que le corresponde según un criterio determinado (cronológico, alfabético o numérico).

Luego de ordenar la documentación se debe realizar la descripción detallada de esta. Con la descripción multinivel se logra la representación del contexto y la estructura jerárquica del fondo y las partes que lo integran. Esta descripción tiene como regla *“en el nivel de fondo proporcionar información sobre éste como un todo. En el nivel siguiente y en los sucesivos, proporcionar información de las partes que se están describiendo. Presenta las descripciones resultantes en una relación jerárquica de la parte al todo procediendo de lo más general, el fondo, a lo más particular”*. [10]

El proceso de describir genera los instrumentos de descripción.

1.1.7. Los instrumentos de descripción

Instrumentos de descripción es un *“término genérico que sirve para denominar cualquier descripción o instrumento de referencia realizado o recibido por un servicio de archivo”*[10].

Los instrumentos de descripción que se generan en un archivo pueden ser tanto textuales como gráficos, una de las clasificaciones que contienen son las guías. Las guías son la clasificación más importante dentro de la gestión documental, describen las secciones de los fondos en dependencia de las necesidades, para lo que están separadas en dos tipos: general y específica. La guía general es el instrumento de consulta que brinda información de los fondos y colecciones que guarda un archivo. Orienta de manera global y destaca las características más importante de un archivo, proporciona información de su origen y creación, de la historia de los organismos productores, ubicación, tipo de servicio, horario, nombre del responsable, entre otros. También se encuentran otras clasificaciones como son los inventarios y los catálogos.

1.2. Los estándares y las normas internacionales

Las normas y estándares internacionales de descripción archivística son reglas para que la información que se distribuye mediante ellas sea entendible por todos, estableciendo un lenguaje común mediante el cual se puedan comunicar todas las instituciones que contengan archivos que sean descritos bajo las mismas.

Establecen pautas estructurales que deben ser respetadas para la correcta representación de la información. Entre las normas que se utilizan se encuentra la Norma Internacional General de Descripción Archivística ISAD(G) y el estándar de Descripción de Archivos Codificados EAD.

Para la comunidad archivística internacional, la utilización de los mismos ha traído consigo que se le preste más importancia a la organización de la información para facilitar su acceso y uso, reconociendo la importancia que tiene una adecuada presentación de los elementos que constituyen la información descriptiva, entre los cuales, unos de los más importantes son los relativos al contexto de producción de los documentos.

Ventajas del uso de normas y estándares para la Descripción Archivística:

- Permite que los archiveros de todo el mundo distribuyan la información sobre sus fondos documentales con facilidad.
- Aseguran una enorme mejora en cuanto a la facilidad de acceso.
- Favorece el intercambio internacional.
- La normalización de registros de autoridad posibilitan la conexión de la descripción de los documentos de archivo con la información relativa a su contexto productivo.
- Posibilita compartir registros de autoridad entre sistemas.

1.2.1. Norma ISAD (G)

A pesar de todos los problemas que han mantenido a la profesión archivística alejada de las normas, la era de la información hace esencial que los archiveros se interesen más en normalizar sus propias prácticas profesionales e influyan en otras normas relativas a la información y en cuánto pudieran afectar la naturaleza e integridad de los archivos.

El hecho de que en cada época y en cada país se desarrolle una técnica y una metodología distinta se debe a múltiples factores como son la variedad y unicidad de los materiales de archivo, tamaño y complejidad de la

organización original, variación en los niveles de descripción y diversidad terminológica entre muchos otros que inciden en la descripción archivística. También *“como en otros procesos archivísticos, la unanimidad no es, ni siquiera aproximadamente, total”* [11].

Para solucionar ese problema una de las normas como reglas generales aplicables a la descripción archivística que fue creada es la Norma Internacional General de Descripción Archivística ISAD(G). Esta *“es una herramienta de gestión muy valiosa para los archivos, brinda especificaciones elementales para la elaboración de descripciones archivísticas compatibles a nivel nacional e internacional, que pueden aplicarse con independencia del tipo documental o soporte físico de los documentos de archivo”*[12].

Está compuesta por un conjunto de reglas generales que son *“parte de un proceso que busca facilitar la integración de las descripciones de diferentes archivos en un sistema unificado de información. Su intención es servir a la elaboración de una primera aproximación a los fondos, en una visión sintética de conjunto”*[12]. Ofrece un mecanismo para utilizar los recursos existentes de una manera más eficiente.

Sus reglas determinan la formulación de la información mediante veinte y seis elementos que pueden ser combinados para constituir la descripción de una entidad archivística, los cuales están clasificados en las siguientes siete áreas de información:

- **Área de identificación:** contiene la información esencial para identificar la unidad de descripción.
- **Área de contexto:** información correspondiente al origen y custodia de la unidad de descripción.
- **Área de contenido y estructura:** información perteneciente al objeto y la organización de la unidad de descripción.
- **Área de condiciones de acceso y uso:** información relativa a la accesibilidad de la unidad de descripción.
- **Área de documentación asociada:** información referente a aquellos documentos que tienen una relación significativa con la unidad de descripción.
- **Área de notas:** información especial y aquella no incluida en ninguna de las demás áreas.

- **Área de control de la descripción:** información de cómo, cuándo y quién ha elaborado la descripción archivística.

De los elementos que forman la norma se consideran seis como esenciales para el intercambio internacional de la información descriptiva:

- Código de referencia.
- Título.
- Productor.
- Fecha(s).
- Extensión de la unidad de descripción.
- Nivel de descripción.

La variabilidad entre los elementos de cada nivel de descripción está solventada mediante la utilización de las denominadas reglas de descripción multinivel anteriormente mencionadas. La descripción multinivel implica que si se va a describir el fondo como un todo, debería representarse en una sola descripción, usando los elementos adecuados de descripción.

1.2.2. Estándar EAD

El estándar de Descripción Archivística Codificada (EAD), ha despertado desde sus inicios la atención de archivistas del mundo entero. La misma constituye “ *el primer estándar de estructura de datos que facilita la distribución de información detallada sobre colecciones y fondos archivísticos, en Internet*”[13]. Además es “*un conjunto de reglas cuya finalidad es determinar las distintas partes intelectuales y físicas de los instrumentos de descripción archivística; con ella se pretende que la información contenida en ellos se pueda consultar, extraer, presentar e intercambiar con independencia de la plataforma que se utilice*”[13].

Pero no son solo estas características la única razón para usar EAD, sino su flexibilidad la cual *”permite codificar todo tipo de documentos y soportes, además de regular la estructura de datos de descripciones archivísticas, proporciona un método de codificación de instrumentos de descripción en cualquier formato, ya sea publicado en Internet, producido por una base de datos o un procesador de texto“*[13].

Entre otras de las características más destacadas de la norma EAD cabe señalar la capacidad de representar la información descriptiva, heredándola de un nivel jerárquico a otro nivel inferior. Además facilita navegar dentro de una estructura jerárquica de información y ayuda en la indexación y recuperación de elementos específicos. No solo los documentos también las imágenes digitales de los documentos se incluyen o se asocian a sus instrumentos de descripción, haciendo posible que los usuarios naveguen por niveles de información más detallados y precisos.

Las Directrices del EAD

Las directrices tienen como objetivo presentar la norma EAD desde diversos puntos de vista, el administrativo, el técnico y sobre todo el archivístico. Son la tercera pieza de la documentación de EAD. No establecen prácticas específicas de codificación (las prácticas descriptivas internacionales actuales son tan divergentes que las reglas estrictas no serían prácticas), sino que ilustran y discuten pros y contras de diversas opciones, tampoco intentan articular un contenido estándar para los instrumentos de descripción.

Están diseñadas para asistir a quienes están considerando usar EAD y quienes están codificando activamente la información descriptiva que se encuentra generalmente en los instrumentos de descripción archivística. Tratan las varias etapas y niveles de implementación de EAD y sus actividades asociadas desde una perspectiva de gestión y de codificación. También pretenden dar perspectivas sobre la base lógica, procesos e implicaciones de la codificación EAD para una audiencia muy amplia, incluyendo administradores, distribuidores de recursos, gestores y supervisores, archiveros y bibliotecarios de referencia, codificadores de instrumentos de descripción, programadores y administradores de sistemas.

Proceso de implementación de EAD

La norma EAD proporciona una estructura estándar para mantener información sobre instrumentos de descripción y sobre las colecciones que describe, pero no normaliza el cómo se determina y cómo se incluye el contenido de cada elemento. La forma óptima para asegurar la consistencia es usar normas descriptivas internacionales. Es por esto, precisamente, por lo que se parte de la descripción utilizando el modelo ISAD (G).

EAD usa el término instrumento de descripción (finding aid) para referirse a cualquier herramienta jerárquica que haya sido codificada usando EAD y que permitirá a un creador de registros o a un usuario acceder a los materiales que están siendo descritos. Permite además la presentación de la información descriptiva de forma extensa e interrelacionada, manteniendo las relaciones jerárquicas entre los distintos niveles de descripción.

En su estructura contiene 146 elementos o etiquetas y 71 atributos para la descripción, entre ellos los 26 elementos que propone ISAD (G). A un nivel muy básico, un documento “instrumento de descripción” codificado utilizando EAD, consta de tres segmentos. El primer segmento que propone, proporciona información sobre el instrumento de descripción en sí mismo (su título, compilador, fecha de compilación) (<eadheader>). Un segundo segmento incluye las cuestiones preliminares necesarias para la publicación formal del instrumento de descripción (<frontmatter>). El tercero proporciona la descripción del material archivístico en sí mismo (<archdesc>), además de la información contextual y administrativa asociada.

- **Cabecera EAD** (EAD Header) <eadheader>: contiene información similar a la de la portada de un texto impreso, pero relativa al documento electrónico, que identifica la versión electrónica del instrumento de descripción y puede también documentar las prácticas de descripción y codificación seguidas en su creación.
- **Material Inicial** (Front Matter) <frontmatter>, opcional: puede usarse para codificar estructuras como prefacios, dedicatorias u otros textos relativos a la creación, publicación o uso del instrumento de descripción, y para generar una portada adaptada a las necesidades o gustos locales.

- **Descripción Archivística** (Archival Description) <archdesc>: representa la descripción archivística propiamente dicha, y puede ser de nivel único (un único fondo, serie, expediente...) o contener una descripción multinivel con la descripción de componentes subordinados (las series de un fondo, las unidades documentales de éstas, en dos o más niveles).

1.3. Estado del Arte

Actualmente a nivel mundial la administración de la información se ha convertido en un recurso de vital importancia económica y organizativa, por lo cual las empresas e instituciones juegan un papel importante en el manejo de la misma. La utilización de estándares mundiales para la descripción archivística ante tanta variedad de criterios traza el camino a seguir para la unificación del conocimiento en normas que definan la forma en que se deben realizar las descripciones.

Mediante la utilización de la ISAD(G) para la descripción de material archivístico se ha garantizado la facilidad de acceso e interconexión entre sistemas de gestión documental, favoreciendo el intercambio de información a nivel nacional e internacional. Si a esto se le suma la convergencia con EAD se multiplica la rentabilidad de los recursos, esto se debe a que EAD normaliza la estructura pero no el contenido. Poder contar con estas dos normas relacionadas entre sí para lograr un propósito, incrementa la posibilidad de explotación de la información descriptiva. Al codificarse los componentes de los instrumentos de descripción se facilita la propia descripción, el control, la navegación, indexación y presentación de la información. El motivo por el cual se utiliza EAD para el manejo de la información descriptiva, es porque facilita el intercambio y la portabilidad, incrementa la inteligibilidad² de los instrumentos de descripción y permite utilizar al máximo las ventajas de las tecnologías de redes.

Entre las aplicaciones existentes en el mundo que utilizan este estándar se encuentra “Spreadsheets to EAD”, la cual es un entorno web que utiliza Ruby para transformar los datos en hojas de cálculo con los niveles de componentes de EAD. La versión actual solo funciona hasta el primer nivel de las series y el segundo nivel de los archivos. En cambio “Ead McTaggart” se encarga hasta seis niveles de componentes y puede ser

²Propiedad de las variedades lingüísticas por la cual dos hablantes de variedades diferentes pueden comprenderse mutuamente

modificada para aumentar su número. Está programada en Visual Basic y permite insertar automáticamente las etiquetas y los atributos necesarios de EAD en una lista de contenedores en Microsoft Excel.

Otra de las herramientas llamada “Yale Finding Aids Creation Tool (FACT)” se utiliza para la creación de instrumentos de descripción, la cual es una versión local personalizada del editor de XML “XMetal Author”. Esta codifica instrumentos de descripción en un esquema local utilizando EAD, además de contar con elementos adicionales incluidos para su visualización durante la edición. También completa los instrumentos de descripción convirtiéndolos en EAD a través de XSLT, incluye hojas de estilo, macros para funciones tales como carpetas y cajas de numeración, creación de enlaces internos e índices, ofrece menús personalizados para adicionar elementos y aplica hojas de estilo sobre los instrumentos de descripción que estén en procesamiento.

En el caso de la aplicación “NCECHO NCEAD 2002” para explorar el patrimonio cultural de Carolina del Norte cuenta con una herramienta que realiza la conversión de instrumentos de descripción codificados con la versión 1.0 hacia EAD 2002. Por otro lado “EAD Web Displays” cuenta con una funcionalidad adicional que permite mostrar el código EAD con secciones plegables, enlaces de hipervínculos a la serie y colores que se alternan en la lista de contenedores para diferenciar los unos de los otros.

La plataforma unificada “Archon” se utiliza para la descripción de archivos y su acceso, posibilita ver, buscar y navegar por la información publicada en un sitio Web, además de seguir la estructura de la norma que permite describir instituciones archivísticas (ISDIAH³).

“Archivists’ Toolkit” es otra aplicación de uso internacional similar a “Archon” de forma interna y ambos tiene como objetivo permitir la importación y exportación de registros EAD, son herramientas de código abierto pero utilizan diferentes licencias. Ofrece apoyo integral a la gestión de la adquisición de materiales de archivo a través de la transformación.

Como se ha podido apreciar son muchas las aplicaciones que operan usando el estándar EAD, pero no todas lo utilizan para importar y exportar agrupaciones documentales y sus descripciones. Las dos últimas aplicaciones mencionadas cuentan con estas funcionalidades, pero son propias de los sistemas a los que están integradas por lo que no se pueden aplicar al eXcriba. Además, a pesar de que son de código abierto requieren

³Norma internacional para describir instituciones que custodian fondos de archivo (International for Describing Institutions with Archival Holdings).

de licencia para su uso.

1.4. Herramientas, tecnologías y metodologías utilizadas

1.4.1. Aplicación de escritorio

Actualmente existen dos tendencias para la implementación de aplicaciones, que pueden ser web o de escritorio, se determinó realizar el módulo como una aplicación de escritorio ya que estas mejoran la experiencia del usuario y son más simples de programar. El costo de desarrollo es mucho menor ya que el programador no tiene que preocuparse por conocer otros lenguajes de programación para dar estilo a las interfaces visuales. No se pierde tiempo de desarrollo haciéndola compatible con los distintos navegadores. Su tiempo de respuesta es más rápido ya que a diferencia de las aplicaciones web la ejecución del código no se realiza en un servidor. A nivel de programación son más robustas y se puede hacer cualquier cosa que permita el sistema operativo. En cuestión de seguridad de la información es mucho más segura, no cualquiera puede acceder a tu aplicación porque no está tan expuesta en comparación a una aplicación web.

1.4.2. Lenguaje de programación

Los lenguajes de programación son aquellos mediante los cuales se controla el comportamiento de las computadoras y otras máquinas. Están constituidos por un conjunto de símbolos, reglas sintácticas y semánticas, que definen su estructura así como el significado de sus elementos y expresiones. Le permite a los desarrolladores dar órdenes de forma precisa.

Los lenguajes de programación se determinan según el nivel de abstracción en lenguajes de nivel bajo, medio, o alto; según la forma de ejecución en lenguajes compilados o interpretados y según el paradigma de programación que poseen cada uno de ellos en lenguajes imperativos, funcionales, lógicos u orientados a objetos.

Teniendo en cuenta que se desarrollará una aplicación de escritorio, se hará una selección de los lenguajes de programación más usados mundialmente con el fin de encontrar el que se adecue a las necesidades del

trabajo.

C++

El lenguaje de programación C++ fue diseñado en los años 80, como extensión del lenguaje C. El C++ mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original.

Las principales características:

- Es un lenguaje que abarca tres paradigmas de la programación: la programación genérica, la programación estructurada y la programación orientada a objetos.
- Es un lenguaje de programación orientado a objetos.
- Es un lenguaje de programación de propósito general, se puede programar desde sistemas operativos y compiladores hasta aplicaciones de bases de datos, procesadores de texto y aplicaciones a medida.
- Es el preferido cuando el rendimiento es fundamental o a la hora de desarrollar aplicaciones de nivel de sistema, controladores de dispositivos o complementos de pantalla ya que hereda estas ventajas de C.
- Es un lenguaje muy versátil y potente.

Java

Java es un lenguaje orientado a objetos desarrollado por Sun Microsystems en los años 90. Este es un lenguaje de programación que tiene un modelo de objetos más simple que C++ aunque sus creadores se basaron en este para crearlo y que elimina herramientas de bajo nivel como punteros, que pueden ocasionar muchas complicaciones.

Características de Java:

- Es un lenguaje orientado a objetos: permite que las aplicaciones creadas con él tengan una visible división en módulos y código reutilizable.
- Es distribuido: diseñado para que la informática distribuida sea más fácil, la programación de redes es como enviar y recibir datos hacia y desde ficheros.

- Es fácil de usar: fue diseñado para ser fácil de aprender y tanto la documentación como los ejemplos son de fácil acceso en internet.
- Toma lo mejor de otros lenguajes: fue basado en C y C++, por lo que toma y mejora las funcionalidades de estos.
- Bajo la licencia GPL: los usuarios no tienen que pagar impuestos sobre patente para usarlo.
- Multiplataforma: permite la ejecución de un mismo programa en múltiples sistemas operativos.
- Posee un recolector de basura que gestiona la memoria y de esta forma optimiza el desarrollo de las aplicaciones, simplificándoles el trabajo. programadores

Se seleccionó el lenguaje de programación java por la posibilidad de que el código pueda ser ejecutado en cualquier plataforma de hardware o software, ya que esta es la característica que favoreció su crecimiento y difusión. Esto se consigue porque java no corre directamente sobre el sistema operativo sino sobre una Máquina Virtual que es creada para cada sistema operativo. Proporciona elevados niveles de seguridad, además es fácil de usar, permite la reutilización de código y no se pagan impuestos de uso.

1.4.3. Entorno de desarrollo integrado

Un entorno de desarrollo (IDE) es un entorno de programación que ha sido empaquetado como una aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI⁴). También pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes, además puede dedicarse en exclusiva a un sólo lenguaje de programación o utilizarse para varios. Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación.

Eclipse

⁴Interfaz gráfica de usuario (del inglés graphical user interface) es un programa informático que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

Eclipse es una comunidad de fuente abierta, cuyos proyectos se centran en la construcción de una plataforma de desarrollo abierta formada por marcos extensibles y herramientas para crear, desplegar y gestionar software en todo el ciclo de vida. Es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Fue desarrollado originalmente por IBM ⁵ y es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

NetBeans

El IDE NetBeans de código abierto es un entorno de desarrollo integrado para los desarrolladores de software. Con su uso se obtienen todas las herramientas necesarias para crear aplicaciones de escritorio, web y aplicaciones móviles en varios lenguajes tales como Java, C / C + +, lenguajes dinámicos como PHP ⁶, Java Script ⁷, Groovy ⁸, y Ruby ⁹. NetBeans es fácil de instalar y utilizar. Por estar implementado en java se puede ejecutar en diferentes plataformas, incluyendo Windows, Linux, Mac OS X y Solaris.

Para la implementación de la aplicación se utilizará el IDE NetBeans, ya que es una de las soluciones más completas para programar en Java. Dispone de soporte para crear interfaces gráficas de forma visual, permite crear indistintamente aplicaciones de escritorio o aplicaciones web y no hay que pagar licencias para su uso.

1.4.4. Metodología de desarrollo de software

Metodología de desarrollo de software en la ingeniería de software es un entorno de trabajo utilizado para estructurar, planificar y controlar el proceso de desarrollo de software.

SXP

SXP es una metodología ágil desarrollada en Cuba que tiene como base para su creación a las metodologías

⁵Empresa multinacional estadounidense que fabrica y comercializa herramientas, programas y servicios informáticos.

⁶Lenguaje de programación interpretado, para la creación principalmente de páginas web dinámicas.

⁷Lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

⁸Lenguaje de programación orientado a objetos implementado sobre la plataforma Java.

⁹Lenguaje de programación interpretado, combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk.

SCRUM ¹⁰ y XP ¹¹ las que permiten el mejoramiento de la producción y que los procesos de desarrollo de software estén actualizados. Está encaminada a aplicársele a proyectos generalmente pequeños con reducidos grupos de trabajo donde los requisitos puedan variar o no estén bien definidos y haya un alto riesgo técnico. Tiene como objetivo la obtención rápida de resultados y una alta flexibilidad. Sus cuatro fases de desarrollo son las siguientes:

- **Planificación-Definición:** Es la fase donde se realiza el aseguramiento del financiamiento del proyecto según las expectativas, dejando bien claro la visión del proyecto.
- **Desarrollo:** Su propósito central como su nombre lo indica es el desarrollo de la implementación del sistema, hasta tenerlo listo para ser entregado.
- **Entrega:** Se le hace entrega del producto al cliente en completo funcionamiento.
- **Mantenimiento:** En esta última fase se realiza el soporte al sistema.

Cada una de las fases antes mencionadas está compuesta por flujos de trabajos y actividades en las que se generan artefactos. Esta es unidireccional en cuanto a trabajo en equipo se trata pues encamina a todo el grupo de trabajo hacia una misma dirección y objetivo a lograr. Permite el chequeo continuo del equipo de trabajo para tener una visión del progreso en las tareas a realizar.

RUP

RUP es una metodología de desarrollo de software la cual es llamada por ese nombre debido a las siglas en inglés de Rational Unified Process. Se caracteriza por dividir el desarrollo de software en cuatro fases:

- **Inicio:** Se establece y determina la visión del proyecto.
- **Elaboración:** Se realiza el análisis del dominio del problema determinando los casos de uso del sistema, se diseña la solución preliminar y se establece la arquitectura en que será desarrollado el producto.

¹⁰Metodología para la gestión y desarrollo de software comúnmente usada en entornos basados en el desarrollo ágil de software.

¹¹Metodologías ágiles llamada Programación Extrema

- Construcción: Se debe de obtener la capacidad operacional inicial.
- Transición: Garantiza que el producto esté preparado para su entrega.

Las fases de RUP son iterativas e incrementales, es por eso que los objetivos para cada una de las iteraciones se establecen en función de la evaluación de las iteraciones precedentes. Cada una de las iteraciones tiene un ciclo de vida que se rige por dos disciplinas.

Disciplina de Desarrollo:

- Modelado del Negocio: Se establecen las necesidades del negocio.
- Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requisitos dentro de la arquitectura del software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte:

- Gestión del cambio y configuraciones: Se guardan todas las versiones del proyecto.
- Administrando el proyecto: Se administran horarios y recursos.
- Ambiente: Se administra el ambiente de desarrollo.
- Distribución: Lograr que el producto sea lanzado con calidad.

Cada iteración que se realice debe de ser clasificada y ordenada según la prioridad que tenga, para convertirse luego en un entregable al cliente. Los entregables constituyen una vía de apoyo para la realización de la siguiente iteración.

Entre los elementos de RUP tenemos:

- Actividades: Son los procesos que se determinan en cada iteración.
- Trabajadores: Entes involucrados en cada proceso.
- Artefacto: Documento o documentación, un modelo o un elemento de modelo.

Esta metodología es más adaptable a proyectos de largo plazo. En cada iteración se hace imprescindible la generación de artefactos.

Se optado por escoger como metodología a usar RUP debido a que se caracteriza por ser iterativo incremental y mejora en cada iteración al producto final, se centra en la arquitectura y se guía por los casos de uso. Tiene una forma disciplinada de asignar tareas y responsabilidades. Administra eficientemente los requisitos y usa una arquitectura basada en componentes, controla los cambios que se realizan en el producto y permite el modelado visual del software. Por todo lo antes expuesto se considera que esta es la metodología más adecuada para el desarrollo del producto.

1.4.5. Lenguaje para modelar los artefactos

Se seleccionó el Lenguaje de Modelado Unificado (UML ¹²) para modelar los artefactos que vayan apareciendo en el proceso de desarrollo del software dado que es un lenguaje consolidado, fácil de aprender, y permite una comunicación fluida entre los diversos actores acerca del modelo. Este lenguaje de modelado es el que se emplea asociado a la metodología de desarrollo que se seleccionó (RUP).

UML es un lenguaje gráfico que se usa para visualizar, especificar, construir y documentar cada parte que compone el desarrollo de software. Con él se pueden modelar conceptos, esquemas de bases de datos, componentes de código reutilizables y describir clases independientemente del lenguaje que se utilice por lo que es un estándar muy utilizado en la notación orientada a objetos. El grupo de diagramas generado por este lenguaje es lo que se conoce por arquitectura del proyecto.

Es desde finales de la década del 90, un lenguaje de modelación orientado a objetos estándar usado de forma preferente por grandes organizaciones como: Rational, Microsoft, Oracle, etc.

¹²Lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

1.4.6. Herramientas CASE

Están formadas por una serie de programas que sirven de ayuda a los ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software, automatizando cada una de sus fases, disminuyendo así el tiempo y el trabajo dedicado a estas. Es la puesta en práctica de conocimientos informáticos en las actividades, las técnicas y las metodologías propias del proceso de desarrollo.

Alguna de las ventajas que brindan las herramientas CASE:

- Posibilitan el aumento de la velocidad en la elaboración de aplicaciones.
- Pueden hacer que analistas y diseñadores ganen tiempo, el cual pueden ocupar en otras tareas.
- Automatizan el modelado de diagramas.
- Complementan la documentación del sistema.
- Ayudan en la creación de base de datos.
- Aprueban la generación de código a partir de los diagramas.

Actualmente estas herramientas junto a una metodología y un lenguaje de modelado, son usadas para en el desarrollo de los software, aumentando la calidad de los diseños de software.

Rational Rose Enterprise

Es una herramienta CASE desarrollada por la International Business Machines (IBM), donde jugaron un papel fundamental los creadores del UML (Booch, Rumbaugh y Jacobson). Está diseñada para brindar soporte al proceso de desarrollo de software. Permite el modelado y la visualización de elementos enmarcados en todo el ciclo de vida del software, lo que favorece la construcción del mismo.

Algunas de las características de Rational Rose Enterprise:

- Utiliza como lenguaje de modelado UML.

- Posee facilidades para la generación de códigos a partir de diagramas previamente elaborados, además facilita la ingeniería inversa de códigos fuentes en distintos lenguajes.
- Basado en el paradigma de programación orientado a objetos.
- Brinda posibilidades de trazabilidad entre los componentes creados durante el desarrollo del software.
- Posee posibilidades para generar documentación automáticamente.
- Es una herramienta propietaria que utiliza como plataforma a Windows.

Enterprise Architect

Herramienta desarrollada por Sparx Systems ¹³, que brinda soporte para facilitar el modelado de procesos y componentes necesarios durante el desarrollo del software, lo que facilita el diseño y la construcción del mismo.

Algunas de las características de Enterprise Architect:

- Es una herramienta orientada a objetos potentes y flexibles.
- Proporciona trazabilidad de componentes durante todas las fases del proceso de desarrollo, creando conectores entre todos los elementos, lo que facilita la organización de estos en diagramas y paquetes.
- Posibilita la generación de código y permite realizar ingeniería inversa en varios lenguajes de programación.
- En las versiones más recientes agrega posibilidades de integración con los entornos de desarrollo Microsoft Visual Studio y Eclipse.
- Es una herramienta para el desarrollo en el sistema operativo Windows.

¹³Compañía informática que radicada en Creswick, Victoria, Australia.

Visual Paradigm

Visual Paradigm para UML es una herramienta que facilita el diseño visual de distintos diagramas. Ayuda en la construcción de los modelos, aumentando al máximo la productividad de los desarrolladores de software.

“Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.”[14]

Algunas de las características de Visual Paradigm:

- Contiene generación de código en varios lenguajes como el caso de Java.
- Para la presentación de los diagramas los puede exportar en formatos jpg o html.
- Posibilita la integración a los principales IDE.
- Posee facilidades para representar especificaciones de casos de uso del sistema.

Visual Paradigm se seleccionó como herramienta para el modelado de diagramas debido a que es herramienta multiplataforma, diseñada tanto para el sistema operativo Windows como para las diferentes distribuciones de Linux. La versión comunitaria ofrece una licencia gratuita. posee una distribución automática de diagramas, ya que cuenta con una organización de las figuras y conectores de los diagramas UML que hace muy fácil el modelado. Permite la exportación de los diagramas hacia imágenes y realizar la ingeniería inversa desde código fuente en Java, C++ y otros.

Conclusiones del capítulo

En el capítulo recién concluido se analizaron los estándares y normas de descripción archivística, así como los principales conceptos relacionados con este tema. Se estudiaron las principales soluciones de software

que contemplan este proceso en la actualidad; permitiendo con ello identificar la necesidad de desarrollar una aplicación que facilite la migración de agrupaciones documentales y sus descripciones entre gestores documentales. De manera adicional, se seleccionaron las herramientas, tecnologías y la metodología necesaria para el correcto desarrollo de la propuesta de solución.

Capítulo 2

Características del sistema

En este capítulo se describirán todos los elementos de la propuesta de desarrollo así como el flujo de todos los procesos involucrados para que se comprendan en su totalidad. Se debe elaborar el modelo de dominio y plasmar los requisitos funcionales y no funcionales necesarios para desarrollar el módulo con los casos de uso correspondientes al mismo.

2.1. Modelo de dominio

En el modelado del negocio es donde se hace un estudio detallado para alcanzar cierto nivel de conocimiento sobre el proceso de negocio que se quiere automatizar, comprender los problemas actuales e identificar las mejoras que se pueden hacer para lograr que los desarrolladores y los consumidores lleguen a un entendimiento común de la organización.

El Modelo de Dominio ayuda a comprender los conceptos que utilizan los usuarios, con los que trabajan y con los que deberá trabajar la aplicación.

Debido a que el nivel de estructuración del modelo de negocio actual es bajo se decide realizar el modelo de dominio el cual ayudará a comprender los conceptos que se centran en la parte del negocio relacionada con el ámbito del proyecto.

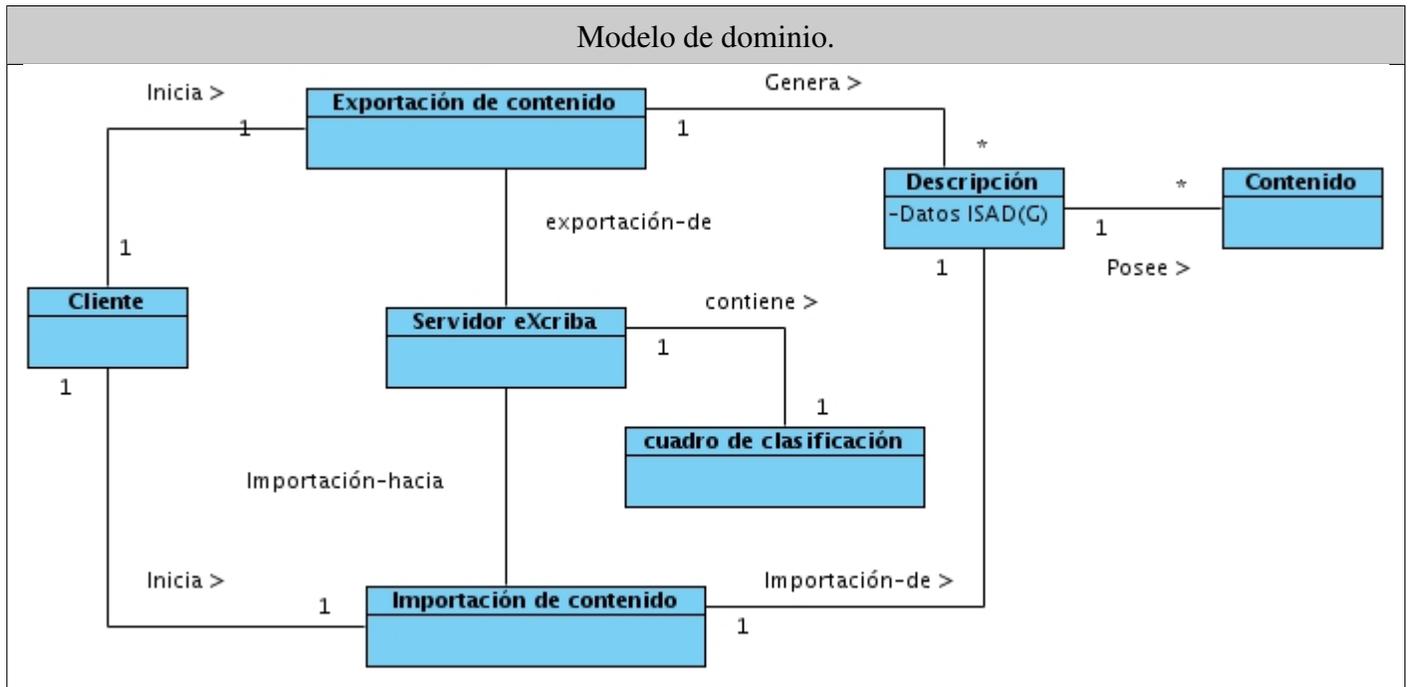


Figura 2.1: **Modelo de dominio.**

- **Cliente:** Persona que interactúa con el sistema.
- **Exportación de contenido:** Acción que se realiza sobre un cuadro de clasificación para realizar alguna salva de información o transportarlo hacia otro sistema.
- **Importación de contenido:** Acción que se realiza sobre un archivo .ead para incorporar al sistema eXcriba metadatos con sus contenidos correspondientes.
- **Servidor eXcriba:** Sistema de gestión documental encargado del manejo de información.
- **Descripción:** Conjunto de campos que contienen valores referentes a contenidos, basados en la norma de descripción documental ISAD(G).
- **Contenido:** Representa cualquier elemento en forma digital, incluyendo archivos, datos, metadatos, así como documentos o sitios web.

- **cuadro de clasificación:** Diagrama jerárquico en el cual se clasifican los documentos.

2.2. Propuesta del sistema

Se desea desarrollar un módulo para eXcriba llamado Enarde que permita la exportación y la importación de agrupaciones documentales y sus descripciones según desee el usuario de manera práctica. Para llevar a cabo esta tarea se implementará una aplicación de escritorio, que le permita al usuario trabajar directamente con el cuadro de clasificación en el servidor de Alfresco, que es el núcleo del sistema eXcriba. Los datos que se exporten quedarán almacenados en un sistema de carpetas con la estructura parecida a la del cuadro de clasificación, acompañado por la descripción en un fichero con extensión “.ead”. El módulo tendrá la posibilidad de funcionar de igual manera en cualquier sistema operativo, esta característica le da una mayor flexibilidad y portabilidad.

Una vez terminado el módulo, debe contener las siguientes funcionalidades:

- **Exportar:** Esta funcionalidad incluye todas las funcionalidades necesarias para exportar las agrupaciones documentales y sus descripciones (descarga el contenido seleccionado y lo organiza de forma jerárquica, luego crea un fichero con la descripción archivística bajo el estándar ISAD (G)).
- **Importar:** Esta funcionalidad incluye todas las funcionalidades necesarias para importar los contenidos hacia el cuadro de clasificación (lee el archivo donde se encuentran las descripciones, con los datos incorpora en el cuadro de clasificación cada contenido donde pertenece y le adiciona la descripción de este).
- **Conectar al servidor:** Permite al usuario establecer una conexión con el núcleo de servidor eXcriba para poder acceder al cuadro de clasificación y leerlo o modificarlo con los nuevos datos.
- **Cerrar la conexión:** Borrar los datos necesarios para acceder al cuadro de clasificación por lo que si se quiere acceder nuevamente se debe establecer una nueva conexión.

2.3. Especificación de requisitos

Según Rogers Pressman *“La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros sistemas software. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si es realizada en forma errónea. Ninguna otra parte es tan difícil de rectificar posteriormente.”* [15]

2.3.1. Requisitos funcionales

Se tuvieron en cuenta los siguientes requerimientos funcionales para identificar que es lo que el sistema debe hacer dado que los requerimientos funcionales son condiciones o capacidades que debe cumplir el sistema.

- **RF 1: Importar contenidos desde un estándar EAD:** Este requisito pretende lograr importar contenidos desde archivos locales a un servidor eXcriba, lo cual permitirá adicionar múltiples contenidos hacia el eXcriba.
- **RF 2: Exportar contenidos en el estándar EAD:** Este requisito pretende exportar contenidos desde un servidor eXcriba bajo el estándar EAD hacia donde el usuario desee, es decir descargar múltiples contenidos desde el eXcriba.
- **RF 3: Autenticarse:** Consiste en una serie de parámetros (servidor, puerto, usuario y clave) que el usuario debe llenar para registrarse en el sistema y lograr la conexión con un servidor eXcriba.
- **RF 4: Navegar por el sistema de ficheros:** permitirá seleccionar los contenidos que se quieren importar desde el disco duro.
- **RF 5: Navegar por el cuadro de clasificación:** Permitirá seleccionar los contenidos a exportar de manera más específica.

- **RF 6: Crear nuevos directorios:** Este requisito pretende que se pueda llevar a cabo una mejor organización de los contenidos exportados.
- **RF 7: Cerrar sesión:** Permitirá cerrar la conexión con el servidor eXscriba una vez concluido el trabajo.

2.3.2. Requisitos no funcionales

Las propiedades o cualidades que el producto debe tener son conocidas como requisitos no funcionales, van ligados a los requisitos funcionales, cuando se sepa que tiene que hacer el sistema entonces se debe saber cómo va a comportarse, cómo se va a ver o con qué rapidez hacerlo entre otras cosas. Los requisitos no funcionales generalmente son fundamentales para el éxito del sistema a construir, pueden marcar la diferencia entre un producto bien aceptado y uno mal aceptado.

Entre las múltiples categorías en las que se clasifican los requisitos no funcionales se tomaron en cuenta para el sistema en cuestión los siguientes:

Requerimientos de software.

Las computadoras que utilizarán el software pueden tener instalado cualquier Sistema Operativo, ya que es multiplataforma. El nodo (PC) que alojará la aplicación deberá tener instalado la Máquina Virtual de Java.

Requerimientos de Apariencia o Interfaz Externa

El sistema debe tener una interfaz con apariencia profesional y un diseño gráfico sencillo.

Requerimientos de usabilidad.

- Diseñar una interfaz amigable, fácil de operar: permitirá que el usuario se sienta a gusto en la aplicación y no tenga que pensar demasiado a la hora de realizar las funcionalidades.
- Utilizar componentes sugerentes a la funcionalidad requerida: el mismo va a contribuir a que el usuario encuentre con facilidad la funcionalidad que busca.
- Correcta distribución de los datos: Los datos estarán organizados de forma lógica para que el usuario no se pierda en el sistema.

Requerimientos de hardware

Las computadoras que utilizarán el software a desarrollar deberán tener 128 MB de Memoria tipo RAM como mínimo. El disco duro debe tener como mínimo 20 GB, dependiendo del tamaño de las agrupaciones documentales a exportar.

2.4. Descripción de los actores del sistema

Los actores del sistema son todos los que interactúan con él, sean artefactos, personas, software o hardware.

Actores.	Justificación.
Cliente	Usuario que va a utilizar el sistema, debe de tener conocimiento de descripción archivística, capacitado para operar en el sistema.

Tabla 2.1: Actor del sistema.

2.5. Diagrama de casos de uso del sistema

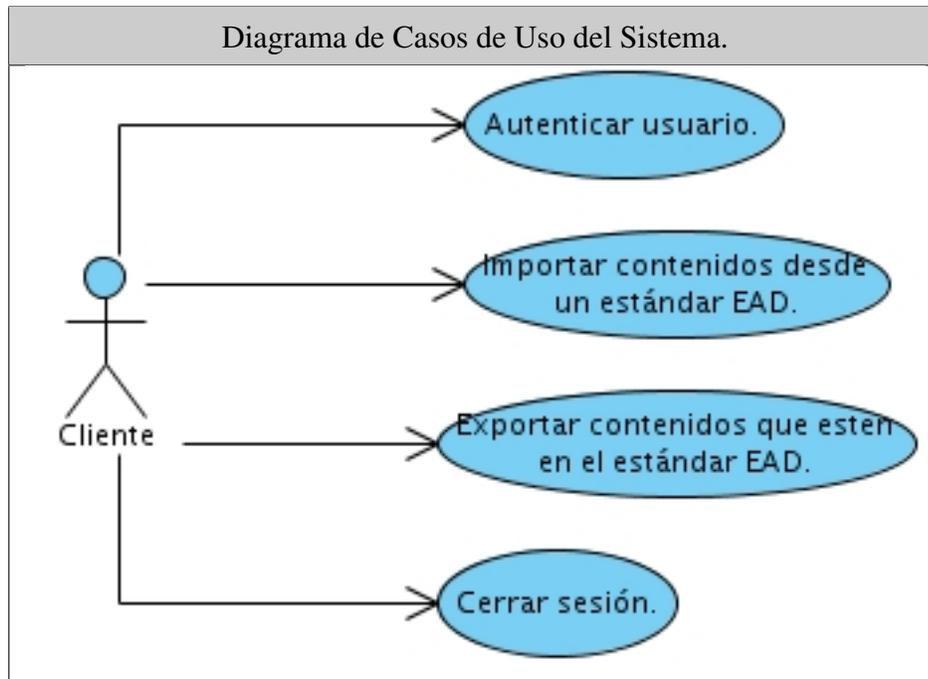


Figura 2.2: Diagrama de Casos de Uso del Sistema.

2.5.1. Definición de los Casos de Uso

A continuación se muestran las descripciones abreviadas de los casos de uso del sistema, las descripciones detalladas se encuentran en el Anexo # 1.

CUS 1: Autenticar usuario.

Caso de uso:	Autenticar usuario.
Actores:	Cliente
Resumen:	El caso de uso inicia cuando el actor decide acceder al sistema introduciendo los parámetros de autenticación, culminando la acción cuándo el sistema le permite el acceso.

Referencia:	RF 3
-------------	------

Tabla 2.2: Descripción del CUS 1 Autenticar usuario.

CUS 2: Cerrar sesión.

Caso de uso:	Cerrar sesión.
Actores:	Cliente
Resumen:	El caso de uso inicia cuando el actor desea salir del sistema, seleccionando la acción Desconectar y finaliza cuando el sistema elimina todos los parámetros de la sesión activa.
Referencia:	RF 7

Tabla 2.3: Descripción del CUS 2 Cerrar sesión.

CUS 3: Importar contenidos desde un estándar EAD.

Caso de uso:	Importar contenidos desde un estándar EAD.
Actores:	Cliente
Resumen:	El caso de uso inicia cuando el actor selecciona la opción Importar, seleccionando la dirección de la descripción EAD y termina cuando el sistema importa todos los datos descritos.

Referencia:	RF 1, RF 4, RF 5
-------------	------------------

Tabla 2.4: **Descripción del CUS 3 Importar contenidos desde un estándar EAD.**

CUS 4: Exportar contenidos en el estándar EAD.

Caso de uso:	Exportar contenidos en el estándar EAD.
Actores:	Cliente
Resumen:	El caso de uso inicia cuando el actor selecciona la opción Exportar, seleccionando la agrupaciones documentales y el lugar donde lo desea guardar. El caso de uso finaliza cuando el nivel junto a su descripción es exportado.
Referencia:	RF 2, RF 4, RF 5, RF 6

Tabla 2.5: **Descripción del CUS 4 Exportar contenidos en el estándar EAD.**

Conclusiones del capítulo

En el presente capítulo teniendo en cuenta las características del sistema eXcriba se analizó la estructura y la dinámica de los procesos involucrados para realizar el modelo conceptual o de dominio. Partiendo de las necesidades que requiere el producto se plantearon un conjunto de requisitos funcionales y no funcionales, así

como los actores que harán uso del sistema. Para especificar la comunicación y el comportamiento del sistema mediante su interacción con estos actores se definieron los casos de uso del sistema.

Capítulo 3

Diseño del subsistema

En el presente capítulo se describe cómo el modulo se realizará; se generan los artefactos que harán posible programar con precisión la solución del problema. Contará con diagramas como el de clases e interacción obtenidos de los casos de usos planteados. Se presentarán los patrones arquitectónicos a utilizar.

3.1. Diseño de la aplicación

El diseño de software es el proceso de definición de la arquitectura, componentes, interfaces y otras características de un sistema. El diseño de Software juega un papel fundamental en el desarrollo de software, permite que el ingeniero de software realice una serie de modelos que forman un plan para la solución de la aplicación y convierte los requisitos en un producto finalizado. En el diseño se hace un refinamiento del análisis y a medida que sigue el refinamiento, el diseño alcanza un nivel de abstracción mucho más bajo.

Sin un diseño o con un diseño mal realizado se tiene el riesgo de construir un sistema inestable, un sistema que puede ser difícil de probar o realizarle cambios, no se podrá evaluar la calidad hasta muy avanzado en el proceso perdiendo mucho tiempo y dinero en él.

El mismo contiene básicamente los diagramas de clases de diseño y los de interacción (secuencia y/o colaboración), conocidos también como las realizaciones de los casos de uso, y finalmente las clases, interfaces y relaciones entre ellas contenidas en los paquetes.

3.1.1. Diagramas de interacción

Se trata de un término genérico para varios diagramas que hacen hincapié en las interacciones entre objetos. Un diagrama de interacción muestra el patrón de interacción entre objetos. Los diagramas de interacción son de diferentes formas, pero todos representan la misma información en ello resaltando cada uno un punto de vista diferente de la información : diagramas de secuencia, diagramas de colaboración.

Diagrama de colaboración

Un diagrama de colaboración muestra una interacción organizada en torno a los objetos que efectúan operaciones. Un diagrama de objetos que muestra los objetos y los enlaces existentes entre ellos que se necesitan para implementar una operación de nivel más elevado. Son una vía alternativa para mostrar un escenario en comparación con el diagrama de secuencia.

Ahora se muestra como quedaron los diagramas de colaboración de la aplicación.

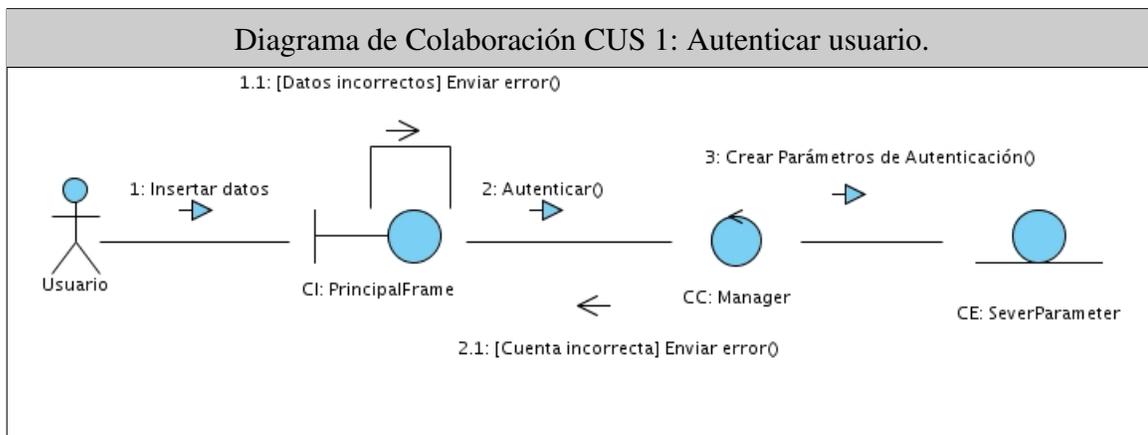


Figura 3.1: Diagrama de Colaboración CUS 1: Autenticar usuario.

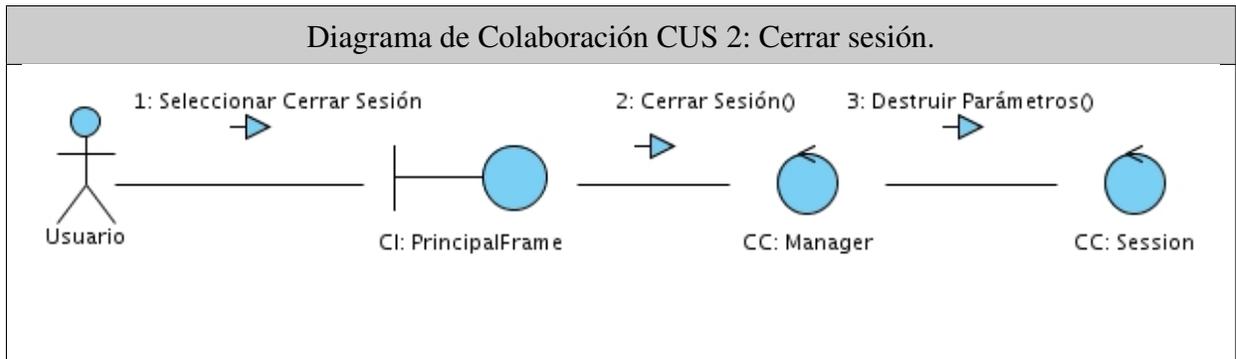


Figura 3.2: **Diagrama de Colaboración CUS 2: Cerrar sesión.**

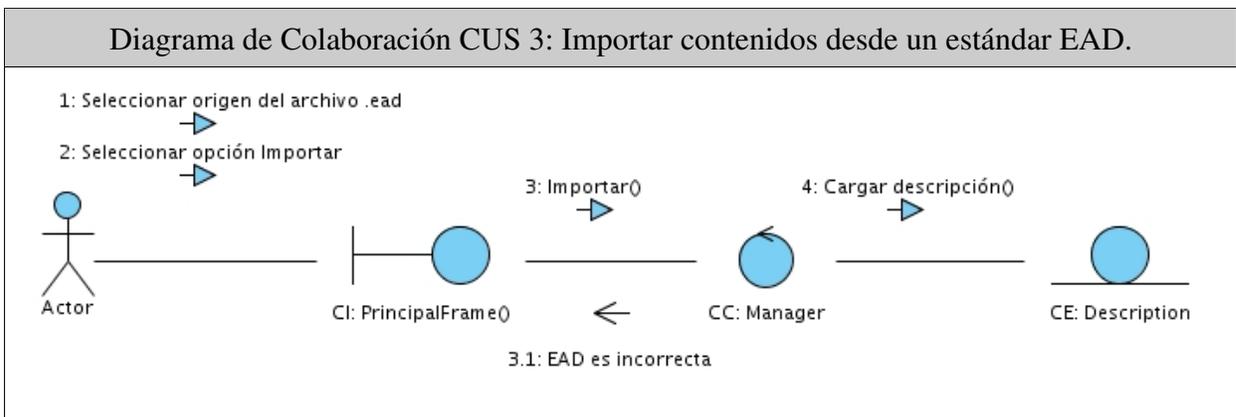


Figura 3.3: **Diagrama de Colaboración CUS 3: Importar contenidos desde un estándar EAD.**

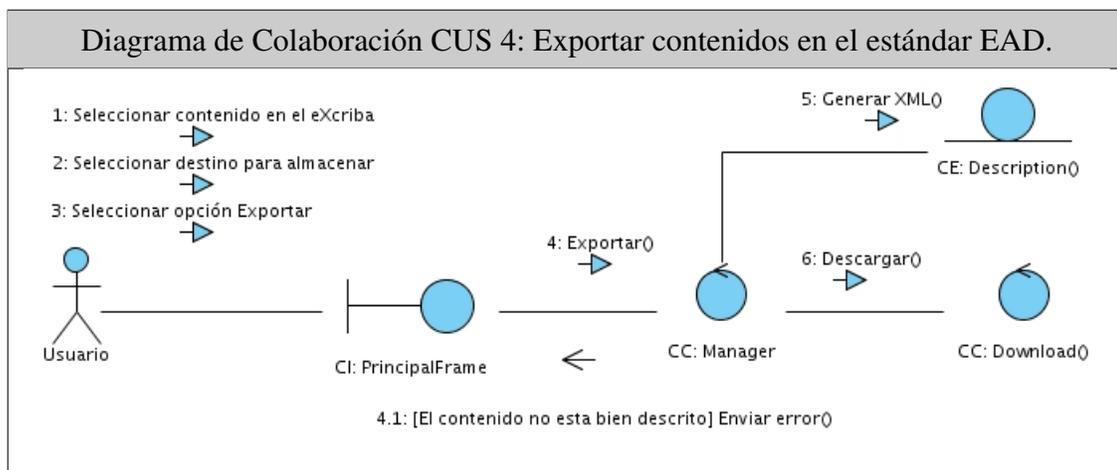


Figura 3.4: **Diagrama de Colaboración CUS 4: Exportar contenidos en el estándar EAD.**

3.1.2. Diagrama de clases del diseño

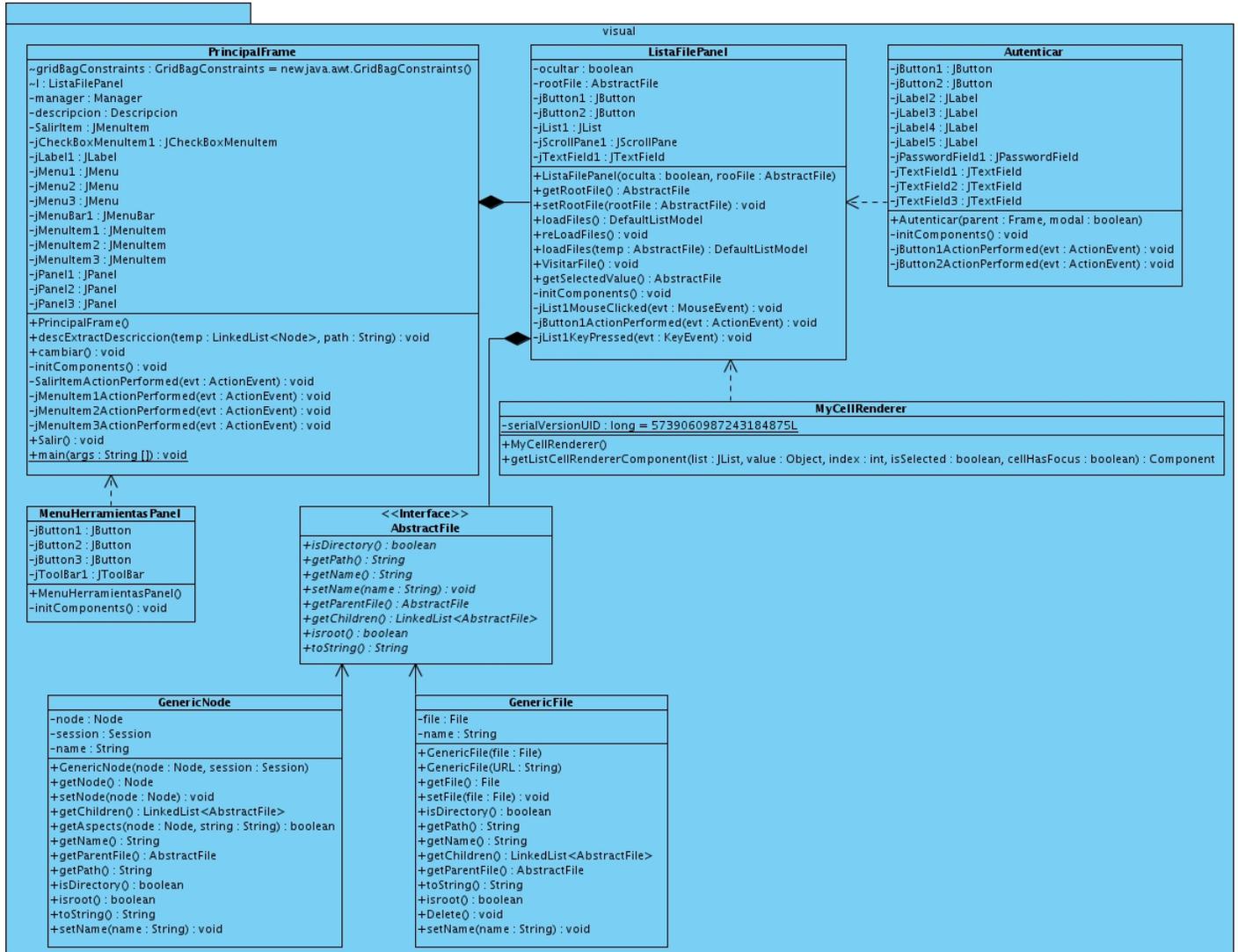


Figura 3.5: Diagrama de clases de diseño del paquete cu.uci.geitel.gda.excriba.enarde.visual.

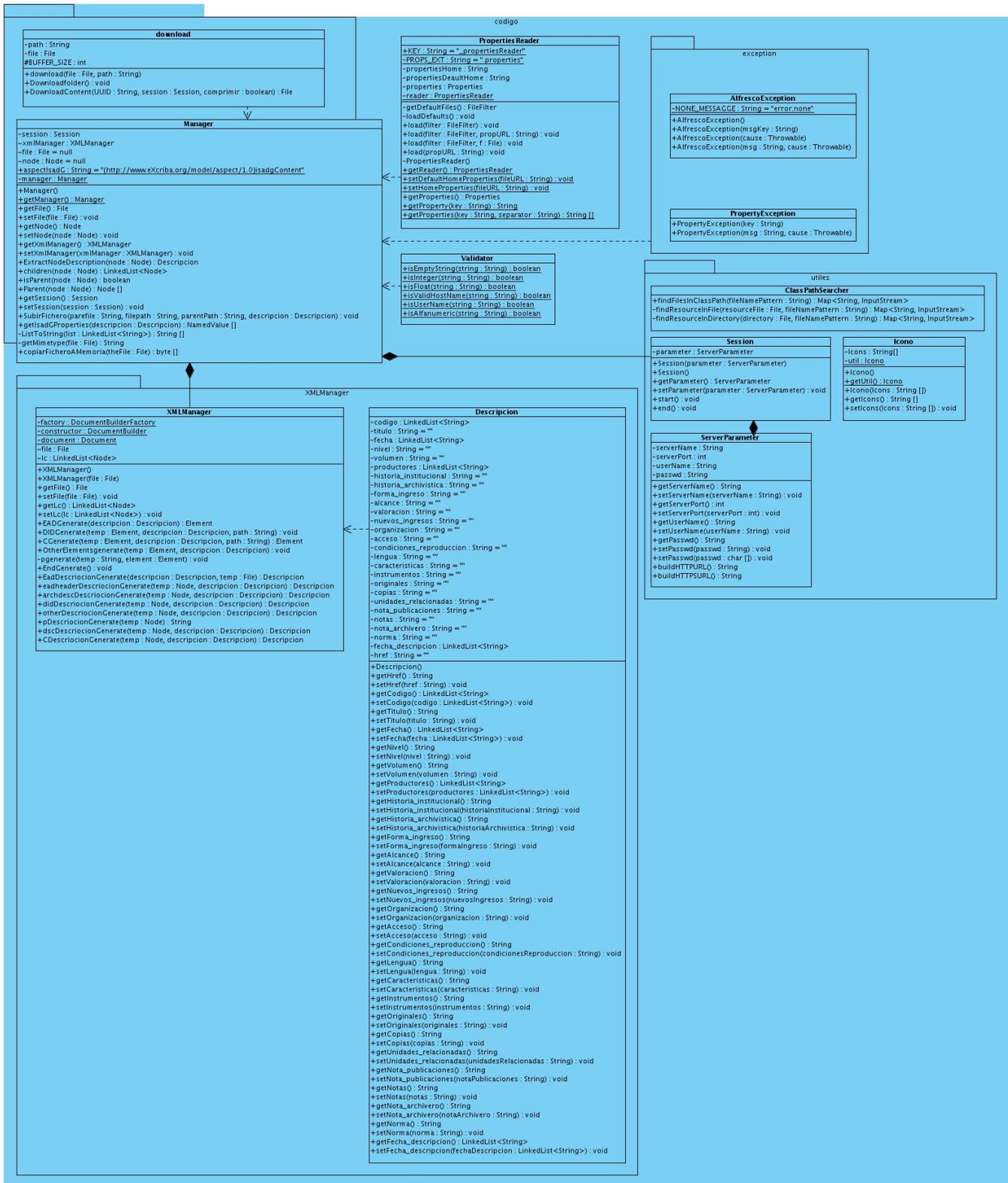


Figura 3.6: Diagrama de clases de diseño del paquete cu.uci.geitel.gda.excriba.enarde.codigo. 43

Descripción de las principales clases

Se realiza la descripción detallada de las clases que conforman los diagramas de clases, especificando los atributos y las operaciones que las mismas realizan.

Nombre: Session	
Tipo de clase: controladora	
Name	Tipo
parameter	ServerParameter
Principales Responsabilidades	
Nombre:	Session(parameter)
Descripción:	Constructor de la clase que iniciará el atributo parameter.
Nombre:	start()
Descripción:	Operación que iniciará la sesión al servidor con los datos almacenados en el atributo.
Nombre:	end()
Descripción:	Operación que finalizará la sesión al servidor.

Tabla 3.1: **Descripción de la clase Session.**

Nombre: ServerParameter	
Tipo de clase: entidad	
Name	Tipo
serverName	String
serverPort	int
userName	String
Passwd	String
Principales Responsabilidades	

Nombre:	buildHTTPURL()
Descripción:	Operación que creará una cadena que contendrá los parámetros necesarios para la autenticación con el servidor por el protocolo “HTTP”.

Tabla 3.2: Descripción de la clase **ServerParameter**.

Nombre: Icono	
Tipo de clase: entidad	
Name	Tipo
Icons	String
util	Icono
Principales Responsabilidades	
Nombre:	Icono(Icons String[])
Descripción:	Constructor de la clase que iniciará una lista de direcciones donde se encuentran los iconos.

Tabla 3.3: Descripción de la clase **Icono**.

Nombre: Download	
Tipo de clase: controladora	
Name	Tipo
path	String
file	File
BUFFER_SIZE	int
Principales Responsabilidades	
Nombre:	Download(file:File, path: String)

Descripción:	Constructor de la clase que iniciará los atributos path con la dirección del contenido a descargar y el file que es donde se pondrá.
Nombre:	downloadfolder()
Descripción:	Operación que se encarga de simular las descargas de las carpetas del servidor creando carpetas en el sistema de ficheros.
Nombre:	downloadContent(UUID String, sesión Session, comprimir boolean)
Descripción:	Operación que se encarga dado un identificador de un contenido de descargarlo del servidor.

Tabla 3.4: **Descripción de la clase Download.**

Nombre: Validator	
Tipo de clase: controladora	
Principales Responsabilidades	
Nombre:	isEmptyString(string String)
Descripción:	Operación que valida que una cadena no esté vacía ni nula.
Nombre:	isInteger(string String)
Descripción:	Operación que valida que una cadena entrada sea un número entero.
Nombre:	IsFoat(string String)
Descripción:	Operación que valida que una cadena entrada sea un número fraccionario.
Nombre:	isValidHostName(string String)
Descripción:	Operación que valida que una cadena entrada sea direcciones en formato nombre o un números IP.
Nombre:	isUserName(string String)

Descripción:	Operación que valida que una cadena entrada sea un nombre de usuario, es decir que comience con letras, luego contenga letras, números o solo los siguientes caracteres válidos (._-).
Nombre:	isAlphanumeric(string String)
Descripción:	Operación que valida que una cadena entrada son contenga números y letras solamente.

Tabla 3.5: Descripción de la clase Validator.

Nombre: Manager	
Tipo de clase: controladora	
Name	Tipo
session	Session
xmlManager	XMLManager
file	File
node	Node
aspectIsadG	String
Manager	Manager
Principales Responsabilidades	
Nombre:	Manager()
Descripción:	Constructor en blanco de la clase.
Nombre:	ExtractNodeDescription(node Node)
Descripción:	Operación que a partir de un nodo extrae la descripción y la devuelve.
Nombre:	children(node Node)
Descripción:	Operación que devuelve una lista de los hijos del nodo pasado por parámetro.
Nombre:	IsParent(node Node)
Descripción:	Operación que devuelve si un nodo tiene hijos.

Nombre:	Parent(node Node)
Descripción:	Operación que devuelve el padre del nodo pasado por parámetro.
Nombre:	unloaderFile(parefile String, filePath string, descripcion Descripcion)
Descripción:	Operación que Sube un fichero hacia el servidor y le adiciona la descripción.
Nombre:	getIsadGproperties(descripción Descripcion)
Descripción:	Operación que Sube un fichero hacia el servidor y le adiciona la descripción.
Nombre:	listToString(list LinkedList)
Descripción:	Operación que concatena una lista de cadenas en una sola cadena.
Nombre:	getMimetype(file File)
Descripción:	Operación que obtiene el tipo de contenido de un fichero.
Nombre:	copyFileToMemory(theFile File)
Descripción:	Operación que carga una archivo en la memoria de la computadora.

Tabla 3.6: Descripción de la clase Manager.

Nombre: XMLManager	
Tipo de clase: controladora	
Name	Tipo
factory	DocumentBuilderFactory
Constructor	DocumentBuilder
Document	Document
file	File
lc	LinkedList<Node>
Principales Responsabilidades	
Nombre:	XMLManager()
Descripción:	Constructor en blanco de la clase.

Nombre:	xmlManager(file File)
Descripción:	Constructor en con un parámetro que inicializa el atributo de donde se va a escribir el XML
Nombre:	eadGenerate(descripcion Descripcion)
Descripción:	Operación que genera las principales etiquetas del XML de tipo EAD a crear.
Nombre:	didGenerate(temp Element, descripcion Descripcion, path String)
Descripción:	Operación genera la etiqueta DID del EAD.
Nombre:	cGenerate(temp Element, descripcion Descripcion, path String)
Descripción:	Operación que genera el próximo nivel en la descripción.
Nombre:	otherElementsGenerate(temp Element, descripcion Descripcion, path String)
Descripción:	Operación genera la parte del EAD en la etiqueta otherelements.
Nombre:	pGenerate(temp String, element Element)
Descripción:	Operación que genera la etiqueta P donde va el contenido de las descripciones.
Nombre:	endGenerate()
Descripción:	Operación que termina con la generación del EAD y lo escribe en el disco.
Nombre:	eadDescriptionGeneration(descripcion Descripcion, temp File)
Descripción:	Operación que lee el fichero EAD y extrae la descripción en las etiquetas principales.
Nombre:	eadHeaderDescriptionGeneration(descripcion Descripcion, temp Node)
Descripción:	Operación que lee el fichero EAD y extrae la descripción en las etiquetas header.
Nombre:	archdescDescriptionGeneration(descripcion Descripcion, temp Node)
Descripción:	Operación que lee el fichero EAD y extrae la descripción en las etiquetas archdesc.
Nombre:	didDescriptionGeneration(descripcion Descripcion, temp Node)

Descripción:	Operación que lee el fichero EAD y extrae la descripción en las etiquetas did.
Nombre:	otherDescriptionGeneration(descripcion Descripcion, temp Node)
Descripción:	Operación que lee el fichero EAD y extrae la descripción en las etiquetas other.
Nombre:	pDescriptionGeneration(descripcion Descripcion, temp Node)
Descripción:	Operación que lee el fichero EAD y extrae la descripción en las etiquetas P.
Nombre:	descDescriptionGeneration(descripcion Descripcion, temp Node)
Descripción:	Operación que lee el fichero EAD y extrae la descripción en las etiquetas desc.
Nombre:	cDescriptionGeneration(descripcion Descripcion, temp Node)
Descripción:	Operación que lee el fichero EAD y extrae la descripción en las etiquetas C que es la que va al próximo nivel.

Tabla 3.7: **Descripción de la clase XMLManager.**

Nombre: Descripcion	
Tipo de clase: entidad	
Name	Tipo
Campos del estándar ISAD(G)	String o LinkenList<String>
Principales Responsabilidades	
Nombre:	Descripcion()
Descripción:	Constructor de la clase.

Tabla 3.8: **Descripción de la clase Descripcion.**

Nombre: PrincipalFrame	
Tipo de clase: interfaz	
Name	Tipo

1	ListaFilePanel
Manager	Manager
descripcion	Descripcion
Principales Responsabilidades	
Nombre:	PrincipalFramer()
Descripción:	Constructor de la clase.
Nombre:	exit()
Descripción:	Cierra la aplicación.
Nombre:	descExtractDescripcion(temp node, path String)
Descripción:	Operación que extrae la descripción desde los nodos.

Tabla 3.9: **Descripción de la clase PrincipalFrame.**

Nombre: ListaFilePanel	
Tipo de clase: interfaz	
Name	Tipo
Ocultar	Boolean
rootFile	AbtractFile
Principales Responsabilidades	
Nombre:	ListaFilePanel (oculta boolean, rootFile AbtractFile)
Descripción:	Constructor de la clase.

Tabla 3.10: **Descripción de la clase PrincipalFrame.**

Conclusiones del capítulo

Una vez finalizado este capítulo se han elaborado los elementos necesarios para dar inicio a la implementación del sistema. Tanto los requisitos funcionales como no funcionales son el basamento principal para diseñar cada funcionalidad con las que debe contar el software. Se recogieron los conceptos manejados, para crear un modelo de objetos que cumplan con las responsabilidades requeridas. Se mostró una vista sobre los aspectos dinámicos del sistema que definen su comportamiento.

Capítulo 4

Implementación y prueba

En el presente capítulo quedará definido cómo será la implementación del sistema, se representará gráficamente los diagramas de despliegue y de componentes, así como las pruebas que se le debe realizar al producto para garantizar su correcto funcionamiento durante todo su ciclo de vida.

Los propósitos de este capítulo serán lo siguientes:

- Definir la organización del código.
- Implementar clases y objetos en forma de componentes.
- Integrar los componentes en un sistema ejecutable.
- Probar los componentes desarrollados.

4.1. Implementación

4.1.1. Diagrama de despliegue

“El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre nodos de computo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del

sistema tiene una influencia principal en su diseño”[14]. Los nodos pueden ser representados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP y otras más como son las microondas.

La figura que se muestra a continuación es el diagrama de despliegue del sistema que se quiere implementar:



Figura 4.1: **Diagrama de Despliegue.**

En la imagen se pueden apreciar la presencia de solo dos nodos, uno de ellos es la PC en la cual se ejecutará la aplicación que se está construyendo y en la otra deberá estar presente el núcleo del servidor eXcriba.

4.1.2. Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre ellos. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables o paquetes. Además de prevalecer en el campo de la arquitectura de software, pueden ser utilizados para modelar y documentar cualquier arquitectura de sistema, tanto la vista estática como dinámica. Muestra la organización y las dependencias entre un conjunto de componentes. No es del todo necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes, cada diagrama describe un apartado del sistema.

Se pueden situar en el mismo librerías, tablas, archivos, ejecutables y documentos que forman parte del sistema. Uno de sus usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

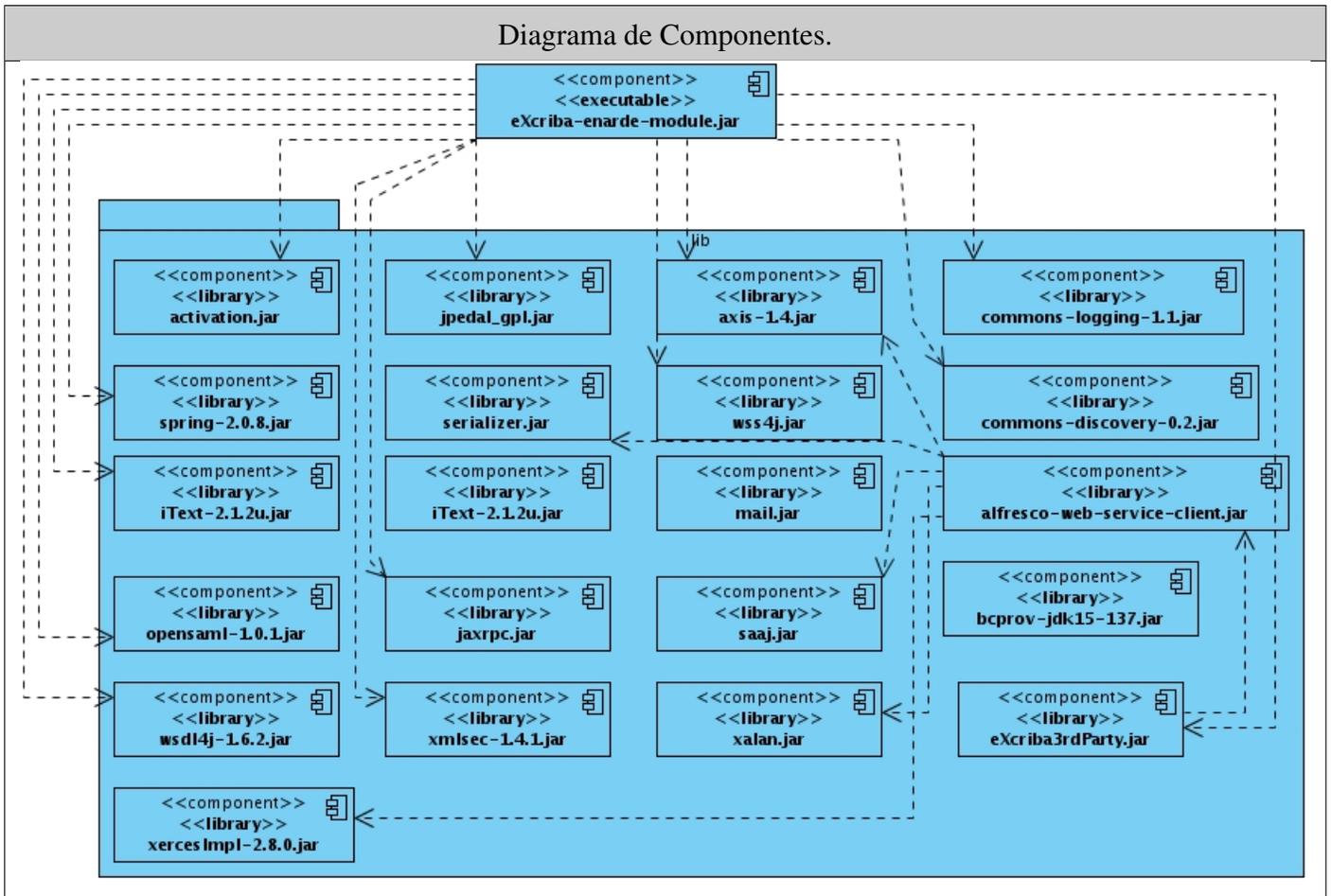


Figura 4.2: Diagrama de Componentes.

4.1.3. Estándar de codificación

Cuando se habla de un estándar de codificación completo este comprende todos los aspectos de la generación de código y debe tender siempre a lo práctico. El código fuente debe reflejar un estilo armonioso, como si hubiese sido escrito por un solo programador de una sola vez.

Al comenzar un proyecto de software debe de establecerse un estándar de codificación para que todos los implicados trabajen de forma coordinada. En caso de que el proyecto incorpore código fuente previo, el estándar debe establecer cómo operar con la base de código existente. La legibilidad del código es de vital

importancia ya que repercute directamente en los programadores con su capacidad de comprender el sistema, en caso de darle mantenimiento reduce los costos en tiempo para añadirle nuevas características, modificar las ya existentes, depurar errores y mejorar el rendimiento. Aunque la legibilidad y la mantenibilidad son el resultado de muchos factores, se le presta especial atención al uso del estándar de codificación como uno de los factores principales que influye en estas dos características.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vista a generar un código de alta calidad es de gran importancia para el producto final del software y como ya se ha mencionado, para obtener un buen rendimiento. Además si se aplica de forma continua un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas, y posteriormente, se efectúan revisiones de código de rutina, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y mantener.

Tipo de identificador	Reglas de nomenclatura	Ejemplos
Constantes	Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un guión bajo (“_”). (Las constantes ANSI se deben evitar, para facilitar la depuración.)	static final int ANCHURA_MINIMA = 4; static final int ANCHURA_MAXIMA = 999;

Paquetes	<p>El prefijo del nombre de un paquete se escribe siempre con letras ASCII en minúsculas, y debe ser uno de los nombres de dominio de alto nivel (actualmente com, edu, gov, mil, net, org) o uno de los códigos ingleses de dos letras que identifican cada país como se especifica en el estándar ISO 3166, 1981. Los siguientes componentes del nombre del paquete variarán de acuerdo a las convenciones de nomenclatura internas de cada organización. Dichas convenciones pueden especificar que algunos nombres de los directorios correspondan a divisiones, departamentos, proyectos o máquinas.</p>	<pre>com.sun.eng com.apple.quicktime.v2 edu.cmu.cs.bovik.cheese</pre>
Clases	<p>Los nombres de las clases deben ser sustantivos. Cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Intente mantener los nombres de las clases simples y descriptivos. Use palabras completas, evite acrónimos y abreviaturas (a no ser que la abreviatura sea mucho más conocida que el nombre completo, como URL o HTML).</p>	<pre>class Cliente; class ImagenAnimada;</pre>
Interfaces	<p>Los nombres de las interfaces siguen la misma regla que las clases.</p>	<pre>Interface ObjetoPersistente; interface Almacen;</pre>

Métodos	<p>Excepto las constantes, todas las instancias y variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres guión bajo “_” y coma “,” aunque ambos están permitidos por el lenguaje. Los nombres de las variables deben ser cortos pero significativos. La elección del nombre de una variable debe ser un mne-mónico, designado para indicar a un observa-dor ocasional su función. Los nombres de varia-bles de un solo carácter se deben evitar, excepto para variables índices temporales. Nombres co-munes para variables temporales son i, j, k, m, y n para enteros; c, d, y e para caracteres.</p>	<pre>int i; char c; float miAn- chura</pre>
---------	---	---

Tabla 4.1: Descripción del estándar de codificación.

4.1.4. Pruebas del sistema

Las pruebas son los procesos que permiten verificar y revelar la calidad de un producto de software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un sistema. Básicamente es una fase en el desarrollo de software que consiste en probar la aplicación construida. Se integran dentro de las diferentes fases del ciclo de la Ingeniería de Software ejecutando un programa que mediante técnicas ex-

perimentales se trata de descubrir los errores que tiene el sistema. Para determinar el nivel de calidad se deben efectuar las medidas o pruebas que permiten comprobar el grado de cumplimiento respecto a las especificaciones iniciales del producto, siendo el resultado observado y registrado. Las pruebas de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y la codificación.

Pruebas de caja negra

El objetivo principal de las pruebas de caja negra es verificar que se se cumplan lo requisitos funcionales planteados. Consiste en estudiar las entradas que recibe y las respuestas que produce determinado sistema, sin tener en cuenta su funcionamiento interno. Solo interesa la forma en que interactúa con el medio que le rodea para entender qué es lo que hace sin importar cómo lo hace. Por tanto deben de estar bien definidas sus entradas y salidas mediante la interfaz del software.

Para verificar el correcto funcionamiento del sistema se deben de crear una serie de juegos de datos los cuales serán las entradas que se le proporcionará al software, estos pueden ser datos formulados correcta o incorrectamente debido a que la aplicación debe de saber manejar cualquier dato introducido sin importar su índole. Una vez introducido cada juego de datos se debe ser muy observador con las respuestas que brinda el sistema para verificar si son los resultados que realmente se esperan.

Entrada	Resultado	Condiciones
Insertar parámetros de autenticación que sean válidos.	El sistema muestra la interfaz correspondiente para acceder a todas las funcionalidades.	Todos los campos a llenar para la autenticación deben de estar correctos.
Insertar los parámetros de autenticación con el campo usuario vacío.	El sistema muestra un mensaje de error “El campo usuario es obligatorio”	El campo que contiene el nombre del usuario no debe estar en blanco y debe contener los caracteres válidos para el nombre de usuario.

Insertar los parámetros de autenticación con el campo contraseña vacío.	El sistema muestra un mensaje de error “El campo contraseña es obligatorio”	La contraseña en el sistema es de carácter obligatorio.
Insertar los parámetros de autenticación con el campo servidor vacío.	El sistema muestra un mensaje de error “El campo servidor es obligatorio”	El campo que contiene la dirección del servidor debe estar presente para poder identificar la ruta en que se encuentra.
Insertar los parámetros de autenticación con el campo puerto vacío.	El sistema muestra un mensaje de error “El campo puerto es obligatorio”	Debe de especificarse el puerto por el cual la aplicación tendrá acceso al servidor.
Insertar en el campo puerto valores incorrectos.	El sistema muestra un mensaje de error “El puerto solo puede ser un valor entero”	Cuando se especifica el número del puerto estos deben ser valores enteros.
Insertar un valor incorrecto en el campo servidor.	El sistema muestra un mensaje de error “La dirección IP del servidor es incorrecta”	La dirección IP del servidor debe estar correctamente elaborada, debe contener la estructura que le corresponde.
Llenar algún parámetro de autenticación con valor que no corresponde con los establecidos para permitir el acceso.	El sistema muestra un mensaje de error “Problemas al establecer conexión con el servidor”	Todos los valores de autenticación deberán coincidir con los que estarán registrados en el servidor, para comprobar si es un usuario tiene permitido el acceso para realizar operaciones en el mismo.

Tabla 4.2: **Caso de prueba Autenticar Usuario.**

Entrada	Resultado	Condiciones
Exportar contenidos a un fichero con extensión ead correctamente.	El sistema exporta el contenido con su descripción correspondiente a un archivo de extensión ead y muestra un mensaje “Proceso finalizado”.	Deberá realizarse los pasos correctos para exportar, estos incluyen seleccionar el/los contenidos a exportar y el lugar en el sistema donde se quieren guardar.
Exportar contenidos que no tengan descrito los campos obligatorios.	El sistema muestra el mensaje de error “El contenido no está bien descrito.”.	Deberán describirse en eXcriba cada nivel del fondo documental que se desea exportar.

Tabla 4.3: Caso de prueba Exportar contenidos en el estándar EAD.

Entrada	Resultado	Condiciones
Importar contenidos desde un estándar EAD correctamente.	El sistema importa el archivo con extensión ead y muestra en la barra de progreso el mensaje “Proceso finalizado”.	Todos los campos a llenar para la autenticación deberán estar correctos.
El fichero *.ead que se selecciono tiene errores de estructura.	El sistema muestra el mensaje de error “El fichero *.ead tiene errores.”	Las descripciones en el fichero *.ead deberán estar estructuradas como especifica en la norma EAD.
No se selecciona un archivo *.ead.	El sistema muestra el mensaje de error “Debe seleccionar un archivo *.ead.”	Para realizar la exportación deberá seleccionarse un fichero *.ead.

Tabla 4.4: Caso de prueba Importar contenidos desde un estándar EAD.

Entrada	Resultado	Condiciones
Elegir la opción desconectar.	El sistema muestra la interfaz de inicio.	El usuario deberá de haberse autenticado previamente.

Tabla 4.5: Caso de prueba Cerrar Sesión.

Conclusiones del capítulo

Con la conclusión de este capítulo se ha establecido una sólida arquitectura en el modelo de despliegue. Se separó el sistema en componentes para mostrar una vista más detallada de las relaciones entre todas sus partes. Y se establecieron pautas que establecen de forma correcta, cómo se debe de trabajar con la codificación del sistema. Una vez concluido el software se le realizaron las pruebas necesarias para verificar su calidad, trayendo consigo luego de su culminación, que se contara con un producto de software listo para el despliegue.

Conclusiones

Con la clausura de este trabajo se cuenta con un producto cuya principal función es facilitar la transportación de datos entre instituciones, mediante la importación y exportación de agrupaciones documentales y sus descripciones entre instituciones.

Se realizó una investigación exhaustiva abarcando los principales conceptos que maneja la gestión documental y archivística para realizar un proceso de desarrollo de software completo. Se aprovechó correctamente las ventajas que brindan las normas y estándares ya existentes en la actualidad, para que todo el proceso llevado a cabo por el módulo se realizara basado en pautas establecidas internacionalmente. Así, con una correcta planificación del tiempo de desarrollo y la capacitación adecuada se logró un producto con la calidad requerida.

La documentación generada como producto paralelo al desarrollo del software quedó recogida en artefactos definidos por la metodología RUP, la cual se usó como secuencia de pasos a seguir durante todo el desarrollo para la satisfacción de los requisitos planteados. Los artefactos servirán como material de apoyo para el posterior entendimiento y estudio de la aplicación.

Recomendaciones

La amplia variedad lingüística presente en todo el mundo, obliga a que las aplicaciones de hoy en día tengan que ser mucho más adaptables y extensibles para que puedan difundirse hacia distintas regiones del planeta. Para que su uso sea mucho más frecuente y por un mayor número de usuarios los cuales puedan acceder a las descripciones, se recomienda:

- Internacionalizar el idioma del software.
- Mostrar las descripciones de los documentos en otros formatos por si se desea hacer un estudio de los mismos.

Para una mayor compatibilidad del módulo con otros sistemas que utilicen otras normas, se recomienda.

- Extender el estudio hacia otros estándares.
- Que la aplicación sea capaz de ser compatible con la norma MARC.

Glosario de términos

UCI Universidad de las Ciencias Informáticas

eXcriba Gestor de Contenido Empresarial de código abierto basado en Alfresco.

SGML Standard Generalized Markup Language. Lenguaje que permite organizar y etiquetar los distintos elementos que componen un documento. Se emplea para manejar grandes documentos que sufren constantes revisiones y se imprimen en distintos formatos de idioma.

EAD Encoding Archival Description. Estándar para codificar las descripciones de los archivos.

W3C World Wide Web Consortium. Consorcio internacional de compañías y organizaciones involucradas en el desarrollo de Internet y en especial de la WWW. Su propósito es desarrollar estándares y "poner orden" en Internet.

XML Extensible Markup Language. Metalenguaje de etiquetado en SGML. Diseñado específicamente para la WWW por W3C. Permite que un usuario diseñe sus propias etiquetas, con sus atributos y reglas de construcción de documentos.

DTD Definición de tipo de documento. Es una descripción de estructura y sintaxis de un documento XML o SGML.

ISAD(G) Norma para estandarizar la estructura de datos, determina cuáles son los elementos que puede contener una descripción, es la guía que se utiliza para llevar a cabo la descripción de un documento.

MARC Formato ampliamente utilizado en las bibliotecas, especifica un estándar en la forma de estructurar la información relativa a los fondos bibliográficos.

UML Es un lenguaje usado para especificar, visualizar y documentar los componentes de un sistema en desarrollo orientado a objetos.

ISIAH Norma que permita describir instituciones archivísticas que guardan documentos de archivo entre otras entidades.

HTML Lenguaje de Marcado de Hipertexto. Es el lenguaje de marcado predominante para la elaboración de páginas web.

CSS Hojas de estilo en cascada Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

Referencias bibliográficas

- [1] “Gestion del conocimiento - gestion documental.” <http://www.gestiondelconocimiento.es/vision.asp>.
- [2] E. G. Huidobro, “Archivólogo.” <http://archivologo.blogcindario.com/2006/07/00259-gestion-documental-conceptos-i.html>, 1998.
- [3] V. H. A. Jordán, “mundoarchivistico.com.” <http://www.mundoarchivistico.com/index.php?menu=diccionario>.
- [4] V. M. Sereno, “Integración de sistemas de gestión electrónica documental en la empresa.” http://www.ciepi.org/fesabid98/Comunicaciones/v_martinez/v_martinez.htm.
- [5] C. I. de Archivos, “Diccionario de terminología archivística.” <http://www.mcu.es/archivos/MC/DTA/Diccionario.html>, 2010.
- [6] M. B. C. ACERO, “Archivos de oficina, secretarias colombia y ecuador, organizacion de documentos en la oficina, cursos virtuales para secretarias.” http://www.archivonorma.com/index.php?secc_id=6&temaidn=20, 2006.
- [7] A. H. Herrera, *ARCHIVÍSTICA GENERAL. TEORÍA Y PRACTICA*. 5ta ed., 1991.
- [8] C. I. D. ARCHIVOS, “Isdiah: Norma internacional para describir instituciones que custodian fondos de archivo,” p. 12, marzo 2008.
- [9] A. Heredia, *Archivística general, teoría y práctica*. 1988.
- [10] A. N. Benlloch, “Isad(g): Norma internacional general de descripción archivística,” p. 19, Sept. 2000.

- [11] P. Eduardo, R. R. Antonio A, and de la Moneda Mercedes, “Propuesta metodológica para la aplicación de la ead encoded archival description : Hacia la información electrónica en archivos.” http://www.ciepi.org/fesabid98/Comunicaciones/e_peis/e_peis.htm.
- [12] “Norma ISAD(G) | archivo histórico.” <http://www.cch.unam.mx/comunicacion/archivoh/capacitacion/isad>.
- [13] G. G. Ruth, “Los metadatos en los documentos electrónicos,” Feb. 2009.
- [14] Jacobson, Ivar, Booch, Grady, Rumbaugh, and James, *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley Publishing Company, 1st.edición ed., Nov. 2000.
- [15] S. P. Roger, *Ingeniería del Software: Un Enfoque Práctico*. McGraw-Hill Companies, quinta edición ed., Feb. 2002.

Bibliografía

- **EAD DTD Version 2002 (EAD Official Site,Library of Congress)** [citado 16 Noviembre 2010]
Disponible en la dirección web:
<<http://www.loc.gov/ead/ead2002a.html>>.
- **Entornos de Desarrollo Integrado para Java** [citado 8 Febrero 2011]
Disponible en la dirección web:
<<http://luauf.com/2008/05/13/entornos-de-desarrollo-integrado-para-java/>>.
- **Glosario de términos** [citado 22 Marzo 2011]
Disponible en la dirección web:
<<http://www.bibliotecalia.com/ayuda/glosario.htm>>.
- **ISAD (G) Norma Internacional General de Descripción Archivística - Archivo General de la Nación** [citado 22 Marzo 2011]
Disponible en la dirección web:
<<http://www.archivogeneral.gov.co/index.php?idcategoria=2216>>.
- **mundearchivistico.com** [citado 16 Noviembre 2010]
Disponible en la dirección web:
<<http://www.mundearchivistico.com/index.php?menu=diccionario&accion=ver&id=162>>.
- **Antonio Eleazaar Serrano López. Modelos de metadatos** [citado 16 Noviembre 2010]
Disponible en la dirección web:

<<http://modelosmetadatos.iespana.es/mestructurales.html>>.

- **Eduardo Peis, Antonio A. Ruiz-Rodríguez. EAD (Encoded Archival Description): Desarrollo, estructura, uso y aplicaciones - (UPF)** [citado 16 Noviembre 2010]

Disponible en la dirección web:

<<http://www.hipertext.net/web/pag223.htm>>.

- **Elisa García-Morales Huidobro. Archivólogo. 1998** [citado 16 Noviembre 2010]

Disponible en la dirección web:

<<http://archivologo.blogcindario.com/2006/07/00259-gestion-documental-conceptos-i.html>>.

- **J. Tomás Nogales. EAD (Encoded Archival Description) Version 2002. Febrero 2007** [citado 16 Noviembre 2010]

Disponible en la dirección web:

<<http://rayuela.uc3m.es/%7Enogales/cursos/ead.html#dtd>>.

- **MARÍA BEATRIZ CANAL ACERO. Archivos de Oficina, Secretarías Colombia y Ecuador, Organización de Documentos en la Oficina, Cursos Virtuales para Secretarías. 2006** [citado 25 Enero 2011]

Disponible en la dirección web:

<http://www.archivonorma.com/index.php?secc_id=6&temaidn=20>.

- **Prof. J. Tomás Nogales Flores. ArchiVóNoMo.biT: 19/07/2007** [citado 16 Noviembre 2010]

Disponible en la dirección web:

<http://urbietorbiarchivonoma.blogspot.com/2007_07_19_archive.html>.

- **Ruth Gilda Gómez. Los Metadatos en los documentos electrónicos** [citado 16 Noviembre 2010]

Disponible en la dirección web:

<<http://www.amigosdelagn.org.ar/53.pdf>>.

- **Society of American Archivists. EAD Application Guidelines for Version 1.0: Home Page. 1999** [citado 16 Noviembre 2010]
Disponible en la dirección web:
<<http://www.loc.gov/ead/ag/aghome.html>>.
- **Ayuda para programación Java: ¿Qué es NetBeans?** [citado 25 Enero 2011]
Disponible en la dirección web:
<<http://ayuda-java.blogspot.com/2007/07/qu-es-netbeans.html>>.
- **Bienvenido a NetBeans y www.netbeans.org, Portal del IDE Java de Código Abierto** [citado 25 Enero 2011]
Disponible en la dirección web:
<http://netbeans.org/index_es.html>.
- **NetBeans** [citado 25 Enero 2011]
Disponible en la dirección web:
<<http://www.worldlingo.com/ma/enwiki/es/NetBeans>>.
- **NetBeans - Wikipedia, la enciclopedia libre** [citado 25 Enero 2011]
Disponible en la dirección web:
<<http://es.wikipedia.org/wiki/NetBeans>>.
- **Encoded Archival Description: An Introduction and Overview** [citado 26 Marzo 2011]
Disponible en la dirección web:
<<http://www.dlib.org/dlib/november99/11pitti.html#5>>.
- **PROPUESTA METODOLÓGICA PARA LA APLICACIÓN DE LA EAD (ENCODED ARCHIVAL DESCRIPTION) : HACIA LA INFORMACIÓN ELECTRÓNICA EN ARCHIVOS** [citado 26 Marzo 2011]

Disponible en la dirección web:

<http://www.ciepi.org/fesabid98/Comunicaciones/e_peis/e_peis.htm>.

- **Alejandro Delgado Gómez - Introducción a Encoded Archival Description (EAD): Mitos y oportunidades** [citado 26 Marzo 2011]

Disponible en la dirección web:

<<http://www.slideshare.net/bunauvarilla/ead1>>.

- Antonia Heredia. Archivística general, teoría y práctica. [Sevilla], 1988.
- Ricardo González Castrillo. Oposiciones a Bibliotecas y Arquivos. 2002.