

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 2**



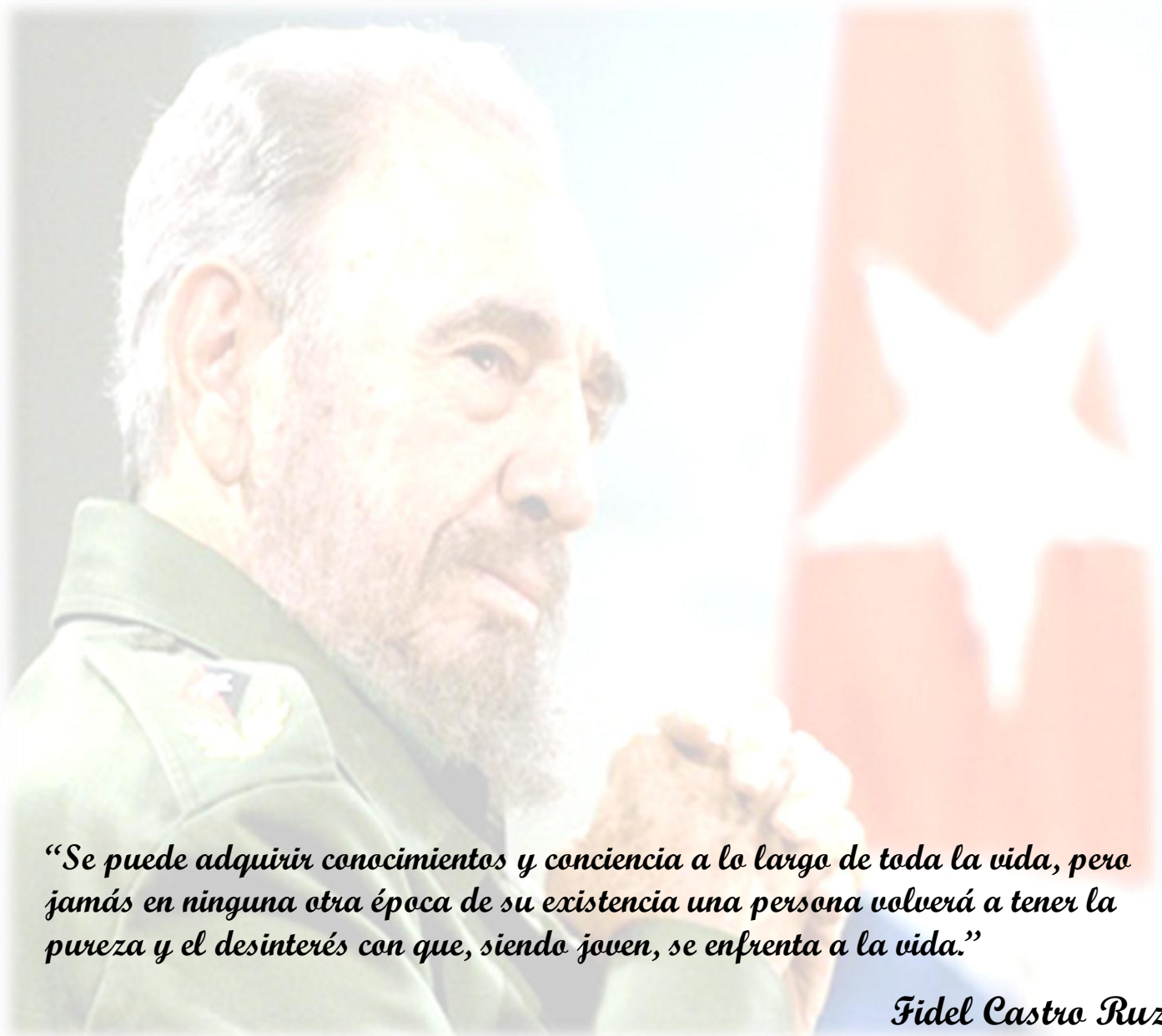
Desarrollo del subsistema Almacenamiento y Conservación del Xabal Arkheia.

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

Autores: Yarelis Abreu González
Yorjander Tan Guevara

Tutor: Ing. Lisbet María González Bravo
Co-Tutor: Ing. Guillermo Enrique Ferrás Pérez

La Habana, Junio del 2013



“Se puede adquirir conocimientos y conciencia a lo largo de toda la vida, pero jamás en ninguna otra época de su existencia una persona volverá a tener la pureza y el desinterés con que, siendo joven, se enfrenta a la vida.”

Fidel Castro Ruz

Declaración de Autoría

Declaración de Autoría

Declaramos ser los legítimos autores del trabajo titulado: “Desarrollo del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia.”, y reconocemos a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

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

Firma del Autor

Yarelis Abreu Gonzales

Firma del Autor

Yorjander Tan Guevara

Firma del Tutor

Ing. Lisbet María González Bravo

Firma del Co-Tutor

Ing. Guillermo Enrique Ferrás Pérez

Dedicatoria

A mi mamá Traida, por ser un ejemplo a seguir de madre y mujer, por querer siempre lo mejor para mí y por ser mi mejor amiga.

A mi papá Elio por estar siempre a mi lado, por darme todo su amor y apoyarme en todo momento.

A mis abuelos Eddy, Araelia y Oreste por brindarme su cariño incondicional.

A mi padrastro Néstor por convertirse en mi segundo padre y preocuparse por mí.

A mi tío Adolfo por complacerme en muchos de mis deseos y siempre estar disponible para mí.

A mi tía Maylin por formar parte de nuestra familia y darme su cariño y bondad.

A mi madrastra Yulie por acompañar a mi padre y darme el regalo más hermoso, mi hermana Yaimelis.

A mi novio Juey por formar parte de su vida, por apoyarme, ayudarme y quererme.

Yarelis Abreu González

Dedico este trabajo a mis abuelos Angela Hernández y Arturo Guevara, o Allín y Pipón, como les llamo desde que nací. Por permanecer siempre a mi lado, brindarme su amor, dedicación y cariño, por sus oportunos consejos, por ser mis guías eternos.

A mi hermana Yusdania, por ser más que mi hermana, ser mi fuente de inspiración a lo largo de mi carrera, este resultado es para ti.

Yorjander Tan Guevara

Agradecimientos

Yarelis Abreu González

Quiero agradecer especialmente a mi familia por brindarme todo el apoyo que he necesitado durante estos 5 años, por sacrificarse y cumplir todos mis deseos.

Agradezco a nuestros tutores Guillermo por brindarnos su apoyo incondicional en todo momento, a Lisbet por convertirse en nuestra compañera y por guiarnos en este trayecto, a los dos por ofrecernos lo más importante su amistad.

A la profesora Madelin Haro por estar dispuesta a ayudarnos cuando lo necesitábamos.

Al papá de mi novio Ivey, la madrastra Ivón, la mamá Lili por estar siempre al tanto de mí y convertirme en un miembro más de su familia.

A mis compañeros del proyecto y tesis Alain, Jany, Rosalia, Deylert por ayudarnos y brindarnos sus conocimientos.

A todas mis amistades en especial a Yaneisy, Irasel, Damaysis, Andy, Ariam, Dany, Adrián, Diannet, Lizzi, Adriana, Alejandro y a mi compañero de tesis Tan, a todos ellos gracias por compartir estos años conmigo y pasar a mi lado los mejores momentos de la universidad.

Yorjander Tan Guevara

A mi tutora Lisbet y a mi co-tutor Guillermo por todo su apoyo e instrucciones, y contribuir con ello, en mi formación profesional, a ustedes agradecerles eternamente.

A la profe Madelin Haro, por ayudarnos a perfeccionar la investigación del trabajo, gracias por enseñarnos a ser mejores profesionales.

A mis abuelos, ustedes son lo más lindo y grande que me ha dado la vida.

A mi mamá Meydis, ma', no tengo palabra para ti, estoy orgulloso de ser tu hijo, este momento también es tuyo.

A mi hermana, por estar presente cuando menos me lo imaginaba y hacerme sonreír con sus ocurrencias.

A mi tío Arturito, por estar siempre que lo necesité. Gracias tío por ser ejemplo en mi vida.

A mi papá Eddy, quizás no has sido lo que un hijo espera, pero al final sé que me quieres.

A mi padrastro Osvaldo, "Tati", por tus consejos y enseñanzas, por brindarme tu cariño de padre.

A mi novia Liz Maviz, por su apoyo, dedicación y cariño, por estar siempre a mi lado.

A mis compañeros Pedro, Arian, Dany, Deylert, Jova y Javier, por ser más que mis amigos, ser mis hermanos. A ustedes les debo mucho.

Agradezco infinitamente a todos mis amigos, profesores y familiares, que influyeron en mí y por apoyarme en todo momento. Agradecer especialmente a Yare por compartir este momento conmigo.

Resumen

Los archivos contienen documentos útiles que perduran durante un largo tiempo, estos ofrecen la posibilidad a la sociedad de consultar e investigar información de interés. Con el objetivo de facilitar el trabajo con los documentos de un archivo y lograr una mejor organización, localización y prolongación de sus vidas se llevan a cabo los procesos de almacenamiento y conservación.

En Cuba prácticamente se trabaja con los documentos de forma manual, lo que conlleva a la pérdida y el deterioro de estos con el transcurso del tiempo. La Oficina del Conservador de la Ciudad de Trinidad y el Valle de los Ingenios no se encuentra exenta de esta situación, por lo que firma un contrato con la Universidad de las Ciencias Informáticas (UCI), para el desarrollo del sistema de gestión de archivos Xabal Arkheia. La aplicación tiene como objetivo agilizar los procesos de organización, incorporación, búsqueda, almacenamiento y conservación de los documentos de los archivos, mediante la informatización de estos. Como parte de Xabal Arkheia se define el subsistema Almacenamiento y Conservación de los archivos, para conocer la ubicación física y posibilitar la prevención de los documentos de los archivos históricos.

El presente trabajo analiza, diseña e implementa el subsistema en cuestión a partir de los requisitos de software definidos y la arquitectura establecida. Se estudiaron las tecnologías, metodología, herramientas y lenguajes para el desarrollo del subsistema. Se realizó un análisis de los sistemas informáticos nacionales e internacionales que reflejó la necesidad de implementar una nueva solución. Para comprobar el funcionamiento del subsistema se aplicaron pruebas de calidad.

Palabras claves: archivo, archivo histórico, documento, almacenamiento y conservación de los documentos de archivos.

Índice

Introducción	1
Capítulo.1 Fundamentación teórica	6
1.1 Introducción.....	6
1.2 Conceptos fundamentales	6
1.3 Análisis de sistemas informáticos.....	9
1.3.1 Sistemas de archivos en el mundo	9
1.3.2 Los sistemas de archivos en Cuba	11
1.3.3 Resultados del análisis de los sistemas.....	13
1.4 Metodologías, Herramientas, Tecnologías y Lenguajes para el desarrollo del subsistema	13
1.4.1 Metodología de desarrollo	14
1.4.2 Herramientas	16
1.4.3 Tecnologías de desarrollo	23
1.4.4 Lenguajes.....	25
1.5 Conclusiones	29
Capítulo.2 Descripción de la solución propuesta.....	31
2.1 Introducción.....	31
2.2 Propuesta de solución	31
2.3 Proceso de negocio del módulo Almacenamiento.....	32
2.4 Proceso de negocio del módulo Conservación	33
2.4.1 Proceso de negocio Conservar Documento.....	33
2.4.2 Proceso de negocio de módulo Conservación para realizar servicio de restauración y/o encuadernación.....	34
2.5 Modelo de dominio.....	34
2.6 Requisitos funcionales	36
2.6.1 Requisitos funcionales del módulo Almacenamiento.....	36
2.6.2 Requisitos funcionales del módulo Conservación	37
2.7 Requisitos no funcionales	38
2.8 Actores del subsistema	39
2.9 Diagrama de casos de uso	40
2.10 Casos de uso.....	41

2.10.1 Casos de uso del módulo Almacenamiento	41
2.10.2 Casos de uso del módulo Conservación	43
2.11 Conclusiones	45
Capítulo.3 Diseño del subsistema.....	46
3.1 Introducción.....	46
3.2 Modelo de Diseño	46
3.2.1 Diagrama de interacción.....	46
3.2.2 Diagrama de paquetes del subsistema	47
3.2.3 Patrón de arquitectura.....	48
3.2.4 Patrones de diseño GRASP.....	50
3.2.5 Diagrama de clases del diseño.....	51
3.2.6 Diagrama de clases persistentes	52
3.5 Conclusiones	55
Capítulo.4 Implementación y Prueba.....	56
4.1 Introducción.....	56
4.2 Implementación	56
4.2.1 Diagrama de componentes	56
4.2.2 Diagrama de despliegue.....	60
4.3 Prueba	61
4.3.1 Pruebas realizadas al subsistema Almacenamiento y Conservación	62
4.3 Conclusiones	65
Conclusiones	66
Recomendaciones	67
Bibliografía	68
Glosario de Términos.....	72

Introducción

La historia de la humanidad está construida a partir de hechos o acontecimientos que tuvieron lugar con el cursar del tiempo. Estos sucesos se conocen debido al privilegio y la precaución de algunos hombres por guardar, conservar y proteger estas acciones. Para recopilar y comunicar información se utilizaron los dibujos, trazados y esgrafiados sobre las rocas o sobre las paredes de las cavernas habitadas por el hombre prehistórico. Se considera que las primeras formas de verdadera escritura fueron dibujos, ideogramas chinos y los jeroglíficos egipcios y aztecas (1).

En la actualidad la información es contenida en documentos, que son guardados en archivos, estos recopilan, conservan y difunden información registrada, generada o recibida por una entidad, grupo o persona en función de las actividades que realizan. Esta información puede estar reflejada sobre papel, cintas o discos de computadora, micro formas, fotografías, fonogramas, videocintas, películas, pinturas, dibujos, mapas.

Para el buen control y resguardo de los archivos existentes en una institución, se trabaja con la archivística. Esta es una disciplina relativamente moderna, conocida como una técnica empírica para el arreglo y conservación de documentos. La archivística permite conocer la ubicación física de los archivos en una institución, en qué estado se encuentran, si están dañados, cómo se pueden arreglar, dónde y cómo pueden estar almacenados.

Para el almacenamiento de los documentos existen estructuras con el fin de ordenarlos físicamente, logrando una mayor organización y localización rápida de la información.

La conservación juega un papel fundamental en la archivística. Garantiza la preservación, duración y el buen estado de los documentos, con el objetivo de que en un futuro puedan existir y brindar un amplio conocimiento a la sociedad.

La conservación es la suma de dos facetas, la preservación y la restauración de los documentos:

- La preservación va dirigida a eliminar daños ambientales o fortuitos que están en el medio que rodean el bien a conservar. Por tanto los métodos y medios preventivos están dirigidos al ambiente, a controlar las condiciones micro climática, con el objetivo de erradicar la degradación de un archivo (2).
- La restauración tiene como fin recuperar la integridad física y funcional de los documentos, gracias a la corrección de las alteraciones que este ha sufrido. (2).

Las Tecnologías de la Información y las Comunicaciones conforman el conjunto de recursos necesarios para manipular, convertir, almacenar, conservar, transmitir y buscar la información; por tales características, es necesaria su aplicación en áreas que requieran manejar y controlar una gran cantidad de información. Este es el caso de los archivos históricos, que para lograr mantener viva la historia de una nación y poder transmitir esta información a futuras generaciones; necesitan almacenar y preservar el estado de la documentación de los fondos archivísticos.

La Universidad de las Ciencias Informáticas es una de las principales instituciones que contribuye al avance de las tecnologías y la informatización del país, formando profesionales en informática y desarrollando software para diferentes entidades. Con el propósito de mejorar la calidad de la gestión de documentos de los archivos, surge el proyecto Archivo, el cual tiene como objetivo el desarrollo del sistema de gestión de archivos Xabal Arkheia para la Oficina del Conservador de la Ciudad de Trinidad y el Valle de los Ingenios. El proceso relacionado con el control de archivos y búsqueda de información en esta institución se realiza manualmente, esto conlleva a una serie de problemas cuando se trabaja con la documentación, como la falta de organización, dificultad para localizarlos, demora para su búsqueda, el deterioro y la pérdida de los documentos.

El gran cúmulo de archivos que existe, contribuye a que se dificulte la adecuada ubicación física de los documentos y que no se cuente con un control estricto de los mismos, lo que obstaculiza el acceso a estos para su consulta. Muchos de los documentos son ejemplares únicos que se deterioran con el tiempo, por lo tanto necesitan de un seguimiento y control según las normas de conservación establecidas. Los documentos en soporte de papel sufren constantes cambios en su composición física y funcional, lo cual pone en peligro la información consignada en ellos, necesitando de un tratamiento específico cada cierto período de tiempo para su conservación.

En función de esta situación, uno de los procesos fundamentales a la hora de implantar el sistema de Xabal Arkheia es el almacenamiento y conservación de los documentos de archivos históricos, para preservar y tener un control estricto de los mismos. Por lo que se propone el siguiente problema a resolver: **¿Cómo facilitar la ubicación física de los documentos y disminuir el deterioro de los mismos, para agilizar los procesos de almacenamiento y conservación de los documentos de los archivos históricos?**

Se define como **objeto de estudio** los sistemas de gestión de documentos en los archivos históricos.

Objetivo General:

Desarrollar el Subsistema Almacenamiento y Conservación del sistema Xabal Arkheia para facilitar la ubicación física de los documentos, disminuir el deterioro de los mismos y agilizar los procesos de almacenamiento y conservación de los documentos de los archivos históricos.

Campo de acción:

Los procesos de almacenamiento y conservación de los documentos de los archivos históricos utilizados por los sistemas de gestión de información.

Objetivos Específicos:

- Confeccionar el marco teórico de la investigación.
- Realizar el diseño del subsistema Almacenamiento y Conservación del Xabal Arkheia a partir de las funcionalidades definidas.
- Desarrollar la implementación del subsistema Almacenamiento y Conservación del Xabal Arkheia en términos de componentes.
- Realizar las pruebas de validación para comprobar el funcionamiento del subsistema Almacenamiento y Conservación del Xabal Arkheia.

Tareas de la investigación:

- Fundamentación del estudio de antecedentes, de las normas y resoluciones que rigen los procesos de almacenamiento y conservación de los archivos en el mundo y en Cuba para realizar el marco teórico del trabajo a presentar.
- Análisis de las funcionalidades de ArchivenHis que sirvan como base para la obtención y descripción de los procesos de negocio.
- Caracterización de la metodología de desarrollo de software, plataforma, lenguaje y el conjunto de herramientas de desarrollo que conforman la línea base de la arquitectura para la elaboración de la aplicación.
- Obtención de los requisitos funcionales y no funcionales de los módulos del subsistema.
- Identificación y selección del patrón de arquitectura y los patrones de diseño más apropiados para la elaboración del producto de software.
- Realización del diseño del software a partir de los requisitos obtenidos.
- Implementación de los elementos de diseño en términos de componentes de implementación.
- Realización de las pruebas al software apoyándose en los diseños de casos de pruebas.

Los métodos científicos utilizados en el presente trabajo son:

Métodos Teóricos:

- **Analítico–sintético:** se utiliza para el análisis de las herramientas, tecnologías, lenguajes y la metodología de desarrollo seleccionadas, identificando las características que las distinguen y haciendo una valoración de porqué fueron escogidas.
- **Modelación:** el lenguaje unificado de modelado (UML) se utiliza para reflejar la estructura, relaciones internas y características de la solución a través de diagramas y para hacer esquemas de cómo deberá quedar la interfaz de usuario del sistema.

Método Empíricos:

- **Observación:** se utiliza para realizar una observación de la información de cada uno de los conceptos relacionados con la gestión de documentos y los procesos de almacenamiento y conservación de los archivos históricos.
- **Entrevista:** se realiza a los especialistas de la Oficina del Conservador de la Ciudad de Trinidad y el Valle de los Ingenios. En esta entrevista se obtiene toda la información posible para la implementación del sistema, procesos del negocio, requisitos funcionales y no funcionales.

El trabajo presenta una introducción, cuatro capítulos, conclusiones, recomendación, bibliografía, referencia bibliográfica y un glosario de términos.

Capítulo.1 Fundamentación teórica. Contiene la fundamentación teórica del tema a desarrollar. Incluye los conceptos fundamentales de la archivística, y los temas principales con los cuales se trabaja en el proceso de almacenamiento y conservación de documentos. Se exponen la metodología, las herramientas, lenguajes y plataformas a utilizar para la implementación del subsistema.

Capítulo.2 Descripción de la solución propuesta: Se describe la solución propuesta del subsistema y su funcionamiento, se exponen los requisitos funcionales y no funcionales del mismo, contiene los diagramas de casos de usos del subsistema Almacenamiento y Conservación del Xabal Arkheia. Se realiza una breve descripción de los actores y los casos de usos de estos módulos.

Capítulo.3 Diseño del subsistema. Tiene como objetivo exponer: los diagramas del diseño, de colaboración de los módulos almacenamiento y conservación del subsistema. Se propone el patrón de

arquitectura a utilizar para el desarrollo del subsistema. Contiene el diagrama de paquete y de clases persistente del subsistema.

Capítulo.4 Implementación y Prueba. Se expone el diagrama de componente y despliegue del subsistema, obteniendo una visión de cómo quedará distribuida la aplicación. Se valida la solución propuesta a través de varias pruebas realizadas al subsistema implementado, para su correcto funcionamiento, calidad y satisfacción del cliente.

Capítulo.1 Fundamentación teórica

1.1 Introducción

En este capítulo se realiza un resumen teórico de los conceptos fundamentales de la gestión de documentos y de los procesos de almacenamiento y conservación de los archivos históricos. Se realiza un análisis de los sistemas informáticos relacionado con la gestión documental, además de una breve descripción de la metodología, las herramientas, tecnologías y los lenguajes, que se utilizan en el desarrollo del sistema.

1.2 Conceptos fundamentales

Archivística

La archivística es la ciencia que estudia la naturaleza de los archivos, los principios de su conservación y organización y los medios para su utilización. Es la disciplina que trata los términos teóricos y prácticos de los archivos y de su función (3).

La archivística surge a partir del siglo XIX, con el nombre de Archivología, su principal objetivo es el arreglo y conservación de los archivos. Es conocida como la ciencia de los archivos que se encarga de la creación, historia y organización de los mismos. Se ocupa del conjunto de archivos integrados en un sistema con toda la problemática que esto comporta (3). Con el tiempo la archivística ha establecido principios esenciales que conllevan a metodologías y lenguajes propios que la identifican de otras disciplinas, ocupando un lugar dentro de la Ciencias de la Documentación e Información.

Existen diferentes tipos de archivística: la “Archivística General”, que contiene los conceptos fundamentales en torno a la disciplina, los documentos, los archivos, sus funciones, las transferencias, organización, descripción, comunicación y difusión de estos. La “Archivística Técnica”, encargada de estudiar la conservación, restauración, reproducción y mecanización de los archivos, y la “Archivística Especial” enmarcada en los diferentes grupos notariales de empresas y en los documentos especiales como los cartográficos, audiovisuales y sellos.

Archivo

El archivo nace cada día, ya sea en el desarrollo de una actividad práctica, jurídica, administrativa de un estado, una ciudad, un grupo organizado, una persona física o de una familia. Es la suma de tres elementos, documentos + organización + servicio.

Documento

Los archivos están conformados por los documentos, pero los documentos no necesariamente tienen que estar en un archivo. Estos integran el patrimonio documental, forman parte de los archivos y de los depósitos documentales, pueden estar sueltos o agruparse en fondos y colecciones.

El documento abarca todo lo que puede transmitir el conocimiento humano como: los libros, revistas, fotografías, filmes, microfilmes, microfichas, láminas, transparencias, diseños, mapas, informes, normas técnicas, patentes, cintas grabadas, discos, partituras, fichas perforadas, manuscritas, sellos, medallas, cuadros y modelos (3).

Documentos de archivo

Los documentos de archivo son la constancia de la información de una oficina pública o privada, de una institución o de una empresa. Este es el testimonio de la actividad del hombre fijado en un soporte perdurable que contiene información (3).

Tipos de archivos y edades de los documentos (4):

Los tipos de archivos son originarios por la teoría de las tres edades de los documentos o el ciclo de vida de los documentos. Esta teoría refleja que los documentos no son algo muerto e inactivo sino que tienen una vida propia. El ciclo de vida de estos es el nacimiento, crecimiento, decrecimiento y la muerte (4).

Etapas o edades de los documentos:

- **La primera etapa o edad**, los documentos se encuentran en fase de tramitación y circulan constantemente. Esta documentación forma parte del “**Archivo Local**”, y en algunos casos, del denominado “**Archivo de Gestión**” y es el archivo de la oficina que reúne su documentación en trámite o sometida a continua utilización y consulta administrativa por las mismas oficinas (4).
- En **la segunda etapa o edad**, la consulta de los documentos es menos frecuente. En este período formará parte del “**Archivo Central**” o “**Archivo Intermedio**”, el cual coordina y controla el funcionamiento de los distintos archivos de gestión y reúne los documentos transferidos por los mismos, una vez finalizado su trámite y cuando su consulta ya no es constante (4).
- En **la tercera etapa o edad**, el documento asume un valor permanente o histórico, restringiéndose su consulta a su carácter cultural e informativo, principalmente con fines de investigación. Su archivamiento y conservación será definitivo en el “**Archivo Histórico**” (4).

En este trabajo se hace énfasis en la tercera etapa de los documentos, es decir, la implementación del subsistema se realiza para los archivos históricos de la Oficina del Conservador de la Ciudad de Trinidad y el Valle de los Ingenios.

Archivo histórico

Los Archivos históricos tienen como finalidad rescatar, clasificar, catalogar, custodiar y poner a disposición los documentos para la consulta pública. Estos constituyen la herramienta básica de un historiador y las pruebas de un pasado que aún puede reinterpretarse a través de nuevas informaciones y documentos que no han sido bien revisados.

Prácticamente la función de un Archivo histórico es la de ordenar, preservar y servir (2):

- Ordenar los documentos de acuerdo a las disposiciones y cuidados pertinentes.
- Preservarlos y cuidarlos de la mejor manera posible, atendiendo las recomendaciones y normas acordes. Dicen que el mayor enemigo de un documento histórico es el hombre mismo.
- Servir como fuente de información, es decir, exponer los archivos para los consultantes y público en general.

Documentos históricos

El documento histórico contiene información que permite conocer el pasado humano (lo que los hombres han pensado, han sentido, han creado o han realizado), bajo el aspecto o ángulo particular según el cual es interrogado (5).

“El documento es, como el lugar de encuentro entre las variables humano-temporales que definen al conocimiento histórico” (5). De lo anterior, se puede decir que no se pasa por la vida sin dejar marcas. Un objeto, una obra, un dibujo, una canción, una carta, una hipótesis formulada son trazos del pasaje del hombre, todo y cualquier huella del pasado, de cualquier naturaleza, define el documento histórico.

Almacenamiento

El almacenamiento es la forma vital de que los archivos estén bien ubicados físicamente, permite una mejor localización y organización de los documentos. Contribuye al control de los archivos de una institución. Corresponde al conjunto de acciones y medidas que deben considerarse y establecerse para garantizar un adecuado acceso, uso, mantención, protección y/o conservación de los documentos de los archivo.

Conservación

La conservación es un conjunto de técnicas, tratamientos y procedimientos que tengan por objeto, tanto la durabilidad y permanencia de los soportes documentales, como la salvaguarda de la información en ellos contenida (2). Para la conservación documental es necesario aplicar la **prevención del deterioro** o sea la preservación del documento y la **reparación del daño**, es decir, la restauración.

Preservación

La preservación previene, estabiliza, neutraliza, controla, y en general, garantiza las condiciones de conservación del patrimonio documental. Sus retos de futuro están en la estandarización de los parámetros de conservación y aspirar a su aplicación tanto para la supervivencia de los soportes como para la supervivencia de la información (6).

Restauración

La restauración cura, repara las lesiones que el tiempo y los hombres por acción u omisión han causado sobre los soportes documentales. Sus retos de futuro están en la industrialización y aplicación masiva de sus técnicas artesanales (6).

1.3 Análisis de sistemas informáticos

La informatización de los sistemas documentales surge para resolver problemas en el almacenamiento y control documental, permitiendo que los procesos dentro de una institución se realicen de forma más eficiente, preservando el estado físico de la documentación. La necesidad de cambiar viejos y rigurosos mecanismos de trabajo en la gestión de documentos y archivos, ha conllevado a la sustitución por disímiles soluciones informáticas capaces de gestionar dichos procesos de forma segura y fácil de manipular.

1.3.1 Sistemas de archivos en el mundo

Existen diversos sistemas para la gestión documental, tal es el caso del sistema de Administración de Documentos (ADOC), el sistema de Gestión Electrónica de los Documentos (DOCUNET), el sistema de Administración Documental de Empresa (SADE.net), el sistema de Solución para la Gestión Documental de Archivos y Correspondencia (SIADOC), y el sistema de Gestión Documental de Archivos (ArchivenHis). Estos sistemas tienen como objetivo principal, facilitar la gestión de documentos de los archivos.

A continuación se realizará una breve reseña de cada uno de ellos.

ADOC LTDA

La empresa de Administración de Documentos ADOC, mediante soluciones de gestión documental, tiene como objetivo lograr que las organizaciones puedan gestionar sus documentos e información en forma digital. Ofrece servicios para digitalizar documentos de archivos históricos o masivos, implementando soluciones de software que permiten realizar procesos de escaneo de documentos, almacenamiento, consulta web y workflow¹ (7). ADOC, provee todos los recursos de hardware, software y humanos para convertir la documentación física en imágenes digitales y utiliza la línea de escáneres KODAK y FUJITSU líderes mundiales en equipos y soluciones para la digitalización de documentos (8). Además, comprende el almacenamiento y administración física de los archivos activos e inactivos. Incluye: Organización, clasificación y depuración documental, Administración de inventarios, Codificación y empaque de documentos, Almacenamiento técnico de cajas de archivo, Servicios de recuperación por distintos medios (9). Entre los sistemas desarrollados por ADOC se encuentra DOCUNET y SADE.Net.

DOCUNET

DOCUNET es un software especializado en la administración y manejo documental y archivístico, adecuado para la conservación de la memoria institucional aplicando organización y normalización a su archivo, basado en los conceptos de gestión documental (10). Se encuentra particionado en tres módulos principales: gestión documental, workflow y el módulo de normas y procedimiento. Entre las características más importantes se encuentran: estructura documental por series, subseries y tipos documentales, digitalización de documentos directa o por lotes, consultas acorde con el perfil de los usuarios, basada en terminología, conceptos y normatividad del Archivo General de la Nación de Colombia, control sobre el préstamo de documentos físicos, uso las bases de datos Oracle, SQL Server (11). Además de algunos requerimientos como la utilización del sistema Operativo Windows. Microprocesador Pentium III o superior. Memoria RAM: mínimo 256 MB y Disco duro: mínimo 10 GB (12).

¹El workflow (**flujo de trabajo**) es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

SADE.Net

Es un sistema de gestión empresarial para la administración integral e inteligente de toda la documentación de la empresa, cubriendo procesos como manejo de archivo, correspondencia, digitalización de documentos físicos, y workflow (7). Entre las características que el sistema posee se encuentran, la administración del archivo físico, transferencias documentales, préstamos, consulta de la documentación de una manera eficiente, rápida y segura, implementación de flujos de trabajo para automatizar procesos documentales, manejo de documentos electrónicos de cualquier formato, uso de las bases de datos SQL u ORACLE (13).

SIADOC

SIADOC es una herramienta para facilitar el control y la administración de sus archivos y expedientes físicos. Compuesto por módulos que permiten la ejecución de las actividades y la administración del archivo de manera organizada. Cuenta con un módulo de administración para su respectiva configuración. Allí se pueden controlar la información del archivo activo e inactivo, control de préstamos, ubicación física, control de modificaciones y actualizaciones. Proporciona un modelo de seguridad por niveles que le permite al usuario gran flexibilidad para controlar el acceso parcial o total del sistema y además, facilita la radicación y gestión electrónica vía Web, de la correspondencia (14). Entre las características y ventajas principales se encuentran: Controlar los documentos y expedientes físicos de la organización y permitir su rápida ubicación. Permitir el acceso a la imagen digitalizada de los expedientes. Administrar el flujo de documentos y su almacenamiento dentro de la organización. Consultar los expedientes de acuerdo a diferentes criterios que facilitan su ubicación y desde cualquier lugar vía Internet o la Intranet de su Empresa. Restringir y controlar el acceso a los archivos. SIADOC está basado en tecnologías de IBM: Lotus Notes/Domino 6.5 o superior (15).

1.3.2 Los sistemas de archivos en Cuba

En Cuba desde el triunfo de la Revolución existe una tradición archivística que data los tiempos de la colonia, orientada a la conservación del patrimonio histórico documental de la nación y a la potenciación de la actividad archivística (16). Desde ese entonces han surgido instituciones y leyes que contribuyen a la protección de los archivos del país.

El Sistema Nacional de Archivos de la República de Cuba

El DECRETO-LEY No. 265 expone algunos artículos relacionados sobre el Sistema Nacional de Archivos de Cuba (17):

ARTICULO 3.

El Sistema Nacional de Archivos de la República de Cuba es el mecanismo de integración y de promoción de la cultura de la gestión documental, que tiene como finalidad lograr el desarrollo armónico de las instituciones o dependencias que lo conforman, para una mayor eficacia en su gestión y en la preservación del Patrimonio Documental de la Nación Cubana, a partir de la aplicación de principios, normas y métodos comunes.

ARTÍCULO 4.

El Sistema Nacional de Archivos de la República de Cuba está integrado por el Archivo Nacional de la República de Cuba y los restantes archivos históricos, centrales, de gestión, especializados, universitarios, así como por los particulares y personales, cuyos titulares así lo decidan.

ARTÍCULO 5.

El Sistema Nacional de Archivos de la República de Cuba funciona sobre la base de la centralización normativa y metodológica y la descentralización operativa.

ARTICULO 6.

Las instituciones y organizaciones donde están ubicados los archivos, tienen la obligación de velar por la integridad de los fondos documentales bajo su custodia y la seguridad de éstos, por la fidelidad de los testimonios que se ofrezcan, así como de garantizar que no se divulgue información que pueda afectar los derechos de las personas jurídicas o naturales. Además, responden por las condiciones materiales que posibilitan la organización, preservación de los archivos a su cargo y por la prestación oportuna y eficiente de los servicios a investigadores, organizaciones y población en general.

ArchivenHis

ArchivenHis es un sistema automatizado de gestión de documentos para un archivo histórico. Permite la conservación y difusión de la documentación existente de un archivo, y efectuar búsqueda de las versiones digitales que existan sobre los documentos. Desarrollado con framework de PHP, cuenta con el gestor de datos MySQL y como entorno de desarrollo el NetBean.

1.3.3 Resultados del análisis de los sistemas

Realizado el análisis a los sistemas anteriormente tratados, se tiene como resultado que existe un patrón genérico en la solución de sistemas archivísticos. Estos están enmarcados en la necesidad de desarrollar el proceso de almacenamiento y conservación de documentos de archivos históricos como parte del sistema Arkheia. Además, de la necesidad de analizar e identificar cuidadosamente los requisitos del sistema para ayudar en la integridad, agilidad, seguridad y gestión del mismo.

El uso de los sistemas analizados (ADOC, DOCUNET, SADE.net, SIADOC), no pueden ser integrados al sistema Xabal Arkheia, ya que son aplicaciones muy generalizadas, con hardware y tecnologías costosas, dependientes de licencias y plataformas privativas. Por su parte ArchivenHis, está desarrollado sobre la base de una tecnología obsoleta, no utiliza ningún gestor documental por lo que el almacenamiento de los documentos se realiza en carpetas, teniendo total dependencia del sistema operativo empleado, además, las funcionalidades que este presenta en cuanto al almacenamiento y conservación de documentos de archivos no cumplen con las necesidades del usuario.

La creación del sistema Xabal Arkheia, viene dado por la necesidad de la administración de archivos, organización, búsqueda y distribución de documentos. Teniendo en cuenta que todo documento atraviesa diferentes estados a lo largo del ciclo de vida desde su creación hasta su caducidad, es importante realizar el almacenamiento y conservación de estos. Como ninguno de los sistemas antes mencionados puede ser utilizado, se hace necesaria la implementación del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia, para preservar el acervo histórico documental.

Con el objetivo de implementar un sistema que cumpla con las necesidades requeridas por la Oficina del Conservador de la Ciudad de Trinidad y el Valle de los Ingenios, se hace necesario contar con un ambiente de trabajo que le posibilite al equipo llevar a cabo el proceso de desarrollo del software.

1.4 Metodologías, Herramientas, Tecnologías y Lenguajes para el desarrollo del subsistema

En el proceso de desarrollo del software es importante definir las metodologías, tecnologías, herramientas de desarrollo, para obtener el producto final, asegurando la calidad del mismo y cumpliendo con las normas establecidas por el grupo de trabajo y el proveedor.

Las metodologías, herramientas, tecnologías y lenguajes de desarrollo que se definieron para la construcción del software, fueron establecidas como políticas del proyecto Sistema de Gestión Documental para la oficina del Conservador de Trinidad y el Valle de los Ingenios, mediante un estudio realizado por el equipo de Arquitectura, por lo que la selección realizada queda fuera del alcance del presente trabajo. Solo se brindará una breve descripción de cada herramienta a utilizar.

1.4.1 Metodología de desarrollo

Las metodologías se definen para dar solución a problemas existentes en la producción de sistemas informáticos. Una metodología es un proceso, que describe los artefactos, roles y actividades asociadas, junto con prácticas y técnicas recomendadas, las características de cada proyecto exigen que el proceso sea configurable, proponiendo procesos diferentes para alcanzar su objetivo, por lo que no existe una metodología de desarrollo de software universal.

Según la filosofía de desarrollo, las metodologías se pueden clasificar en:

- Metodologías tradicionales o robustas: Se basan en la planificación y se aplican a proyectos de gran tamaño respecto a tiempo y recursos. Estas no ofrecen una buena solución para proyectos donde el entorno es volátil y donde los requisitos no se conocen con exactitud, porque no están pensadas para trabajar con incertidumbre. Son menos orientadas al código y dedican más tiempo pensando en cómo se debe desarrollar el sistema que en programar el desarrollo y las pruebas.
- Metodologías ágiles: Se centran en el software y no en su diseño y documentación, dirigidas sobre todo a equipos de desarrollo pequeños, asociados al trabajo en equipo e involucran al cliente en el proceso como parte activa del propio equipo de desarrollo. Fueron diseñadas principalmente para apoyar el desarrollo de aplicaciones de negocio, donde los requerimientos cambian constantemente y están pensados para entregar de forma rápida y convencional el software a los clientes.

Proceso Unificado de Racional (RUP)

La metodología de desarrollo Proceso Unificado de Racional (RUP) por sus siglas en inglés (Rational Unified Process), es un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo, siendo su meta asegurar la producción del software con alta calidad resolviendo las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos. RUP define un marco de trabajo genérico especializado

para una gran variedad de sistemas software, siendo factible para el desarrollo de soluciones informáticas complejas y extensas en cronogramas de ejecución, producto de las características que posee.

RUP posee dos dimensiones (Figura.1) (18).

- La primera dimensión representa el aspecto dinámico del proceso y se expresa en términos de fases, de iteraciones, y la finalización de las fases.
- La segunda dimensión representa el aspecto estático del proceso: cómo se describe en términos de componentes de proceso, las disciplinas, las actividades, los flujos de trabajo, los artefactos, y los roles.

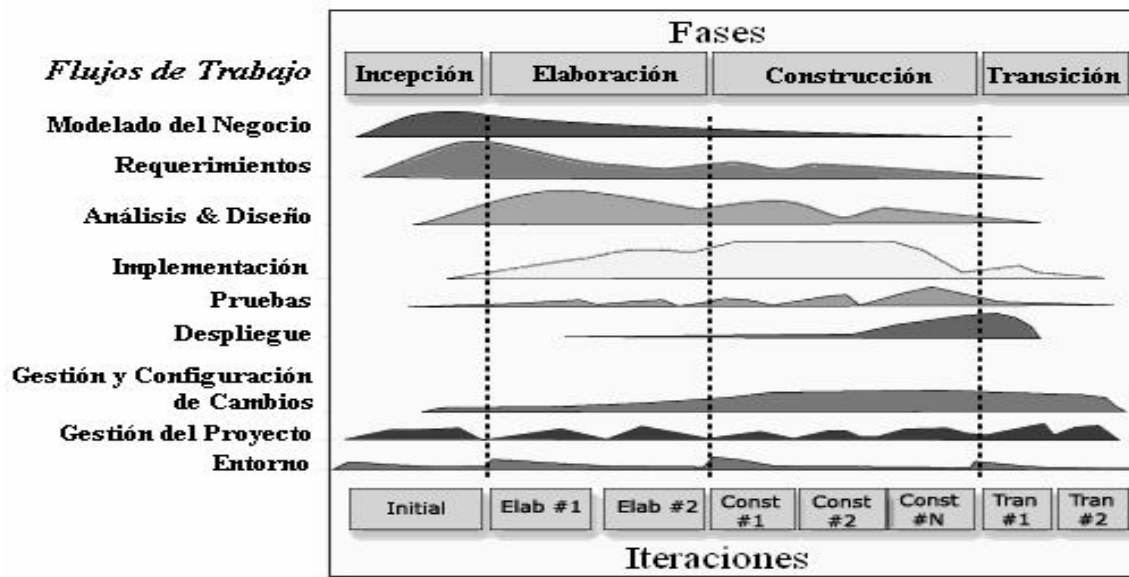


Figura.1 (Disciplinas, fases, iteraciones del RUP).

RUP posee tres características esenciales que lo distingue:

- Dirigido por los Casos de Uso: Los casos de uso se utilizan para describir los requisitos funcionales del sistema desde la perspectiva del usuario, estableciendo el comportamiento que se desee del sistema, para validar y verificar la arquitectura definida en el sistema así como para determinar el alcance de cada iteración y el contenido de trabajo de los integrantes del equipo de desarrollo.
- Centrado en la arquitectura: La arquitectura del sistema permite ganar control sobre el proyecto para manejar su complejidad y controlar su integridad, mostrando una visión común del sistema

para los clientes y el equipo de proyecto. Se utiliza para construir, conceptualizar y evolucionar el sistema. Hace posible la reutilización a gran escala y provee una base para la gestión del proyecto.

- Iterativo e incremental: El proceso iterativo e incremental consta de una secuencia de iteraciones a partes pequeñas o mini-proyectos siendo más práctico. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto (18).

Existe relación entre las tres características, donde la arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración.

Este proceso RUP lo divide en 4 fases: Inicio, Elaboración, Construcción y Transición, siendo divididas cada una de ellas en iteraciones. En cada iteración se trabaja en un número de disciplinas haciendo énfasis en algunas de ellas. Las disciplinas definidas por RUP son: Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, entre otras.

RUP es una metodología de desarrollo estándar que se utiliza en el análisis, implementación y documentación de sistemas de desarrollos que se adapta a la necesidad de cada organización. Atendiendo a las características que proponen las metodologías tradicionales, las particularidades y necesidades que presenta el proyecto y su equipo de desarrollo, se selecciona a RUP como metodología de desarrollo para el sistema informático Xabal Arkheia. Mediante el cual se podrá lograr una mejor organización y abundante documentación, posibilitando que el avance del proyecto no se vea afectado y asegure su continuidad, atendiendo que el equipo de desarrollo es grande e inestable, compuesto por estudiantes y profesores que pueden cumplir otras funciones dentro del mismo.

1.4.2 Herramientas

1.4.2.1 Herramientas de Modelado

Las herramientas CASE (Ingeniería de Software Asistida por Computadora), se puede definir como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida del desarrollo de un software. CASE es también

definido como el conjunto de métodos, utilidades y técnicas que facilitan el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

Visual Paradigm For UML 8.0

La herramienta está diseñada para una amplia gama de usuarios, que incluye desde los ingenieros de software hasta cualquier persona que esté interesada en la construcción de forma fiable de sistemas de software con un enfoque orientado a objetos.

Tiene como características que es un producto de calidad, soporta aplicaciones web, presenta generación de código para Java y exporta como HTML. Es fácil de instalar y actualizar y tiene compatibilidad entre ediciones, propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Actualmente es muy utilizada internacionalmente debido a su licencia gratuita y comercial. Esta herramienta de modelado ha sido concebida para un ciclo completo (análisis y diseño, construcción, pruebas y despliegue) del proceso de desarrollo del software, es de gran utilidad para el trabajo del analista.

El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Es una herramienta de modelado visual para todos los tipos de diagramas UML. Es compatible con una amplia gestión de casos de uso, lenguaje de especificación de sistemas para diagramas de requisitos y diseño de base de datos (19). Es además una herramienta que facilita su asimilación y entendimiento por parte del equipo de desarrollo, minimizando el tiempo en el desarrollo de la arquitectura del sistema. El Visual Paradigm permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML (20).

Dentro de sus Características se encuentra (20):

- Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OSX, Solaris o Java.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Generación de código para Java y exportación como HTML.
- Generación de bases de datos. Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Sincronización entre diagramas.
- Generación de documentos.
- Integración con distintos Ambientes de Desarrollo Integrados (IDE).

Se selecciona el Visual Paradigm como herramienta Case por su integración al Lenguaje Unificado de Modelado (UML), por ser multiplataforma, amigable en su uso y poseer interoperabilidad con otras aplicaciones e integrarse con distintos IDE. También por sus disímiles características ya mencionadas con anterioridad. UML, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, que ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables (20).

1.4.2.2 Herramientas de Desarrollo

Las herramientas informáticas son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillo, en otro sentido se puede decir que una herramienta es cualquier programa o instrucción que facilita una tarea, donde cada herramienta se crea y diseña para una o varias funciones determinadas (21).

SpringSource Tool Suite (STS) 2.5

Es un entorno de desarrollo integrado multiplataforma de código abierto desarrollado por SpringSource que soporta una amplia gama de lenguajes de programación (22). Posee soporte para integrar el contenedor web Apache Tomcat para el despliegue de aplicaciones web. SpringSource también brinda la posibilidad de lanzar comandos Grails y posee un soporte de depuración mejorado, así como mejoras en el tipo de inferencia y soluciones rápidas en el editor de Groovy.

STS es una herramienta libre, destinada a construir aplicaciones empresariales enriquecidas por los proyectos de SpringSource, no sólo incluye herramientas para el desarrollo en lenguaje Java, sino también, para Groovy, Grails y otros lenguajes Spring (22).

El uso de esta herramienta prevé de posibles cambios o migración de lenguaje o sistema operativo, los cuales no afectaría al equipo de desarrollo. Además de facilidades que brinda atendiendo a las características antes mencionadas, es un entorno de desarrollo amigable, entendible y potente para el uso del lenguaje que se utilizará en el desarrollo del subsistema.

TortoiseSVN 1.7

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones Apache Subversion. Esto significa que TortoiseSVN administra archivos y directorios a lo largo del tiempo. Los archivos se almacenan en un repositorio central. El repositorio es prácticamente lo mismo que un servidor de archivos ordinario, con la excepción de que recuerda todos los cambios que se hayan hecho a sus archivos y directorios (23).

Algunas de las características de esta herramienta son (23):

- Integración con la consola de Windows: TortoiseSVN se integra perfectamente con la consola de Windows. Lo que mejora el trabajo del usuario permitiendo continuar el trabajo con las herramientas sin cambiar de aplicación.
- Íconos superpuestos: El estado de cada carpeta y archivo versionado se indica por pequeños íconos superpuestos. De esta forma, puede conocer fácilmente el estado en el que se encuentra su copia de trabajo.
- Fácil de usar: Todos los comandos están disponibles directamente desde Windows Explorer, muestra sólo los comandos que tienen sentido para el archivo ó carpeta seleccionada, los cuadros de diálogo descriptivos son constantemente mejorados debido a los comentarios de los usuarios y permite mover los archivos arrastrándolos justo en Windows Explorer.
- Interfaz Gráfica de Usuario: El usuario puede ver los cambios realizados a un archivo o carpeta.

Se selecciona esta herramienta de trabajo, teniendo en cuenta que además de las características antes mencionadas, predomina la importancia de su uso en el desarrollo del sistema. El TortoiseSVN fomenta la colaboración del equipo de trabajo, dado que varios usuarios pueden modificar y administrar el mismo conjunto de datos desde su ubicación. Teniendo en cuenta que la herramienta puede crear repositorios de ficheros, los cuales pueden ser accedidos mediante la red desde distintos ordenadores.

1.4.2.3 Gestor de Base de Datos

Un sistema gestor de base de datos tiene como objetivo simplificar y facilitar el acceso a los datos y hacer que los tiempos de respuesta a las solicitudes de los usuarios sean muy reducidos. Un sistema de gestión de bases de datos está constituido por un conjunto de datos interrelacionados y programas para acceder a estos datos.

PostgreSQL 9.1

Es un sistema de gestión de bases de datos objeto-relacional, de propósito general, multiusuario y de código abierto distribuido bajo licencia BSD (Berkeley Software Distribution), soporta gran parte del estándar SQL y ofrece modernas características como consultas complejas, disparadores, vistas, integridad transaccional, control de concurrencia multiversión.

PostgreSQL 9.1 dado sus características técnicas que se han venido desarrollando como la estabilidad, potencia, robustez, facilidad de administración e implementación de estándares la hacen una de las bases de datos más potentes, y robustas del mercado. PostgreSQL funciona correctamente con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

Algunas de sus características son: (24)

- Sistema de gestión de bases de datos objeto-relacional: capaz de manejar rutinas complejas y reglas.
- Garantiza la integridad referencial: utilizada para garantizar la validez de los datos de las bases de datos.
- Tiene API (Application Programming Interface) flexible: lo que permite el soporte para un desarrollo más fácil y de varios lenguajes.
- Tiene una arquitectura cliente/servidor: arquitectura de proceso por usuario y cliente/servidor; hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectarse al gestor.
- Multiplataforma: soporta alrededor de 34 plataformas incluyendo Linux y Unix en todas sus variantes y Windows.
- Posee herramientas gráficas de diseño y administración de bases de datos: existen varias herramientas gráficas de alta calidad para administrar las bases de datos como el pgAdmin.

- Soporta bloques de código: que se ejecutan en el servidor y que pueden ser escritos en diferentes lenguajes de programación con la potencia que presenta cada uno.
- Replicación Sincrónica: permite alta disponibilidad con consistencia sobre múltiples servidores.

Fue seleccionado PostgreSQL debido a que es un gestor de base de datos multiplataforma, presentando una licencia liberal, la que puede usarse, modificarse, y distribuirse gratis para cualquier propósito, presentando una alta estabilidad y confiabilidad. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL es manejado por una comunidad de desarrolladores que trabajan de forma desinteresada, libre y apoyada por organizaciones comerciales, lo cual emite abundante documentación y soluciones. Además se tuvo en cuenta para su selección el proceso de transición y migración hacia software libre en el que se encuentra el país.

1.4.2.4 Gestor Documental

Nuxeo

Nuxeo es una plataforma de código abierto para gestión de contenidos empresariales, desarrollado en Java. Es extensible, configurable y modular para que las aplicaciones se puedan configurar y personalizar para adaptarse al contexto del proyecto. Viene con módulos funcionales para responder a necesidades genéricas, tales como la gestión de documentos para la gestión de contenidos, colaboración social para la gestión de contenidos centrados en el usuario y el proyecto, gestión de activos digitales para la gestión de contenido de medios ricos (incluyendo imágenes, audio y vídeo) y flujo de trabajo de gestión de casos para la gestión de casos utilizado tal cual o como base para aplicaciones más específicas del negocio (25).

Algunas de sus características son:

- Gestión de documentos, contenido web, registros e imágenes.
- Publicación integrada.
- Soporte de varios idiomas.
- Soporte multiplataforma (Windows, Linux, Solaris, Mac OS).
- Interfaz gráfica basada en navegadores de Internet.

- Flujo de trabajo basado en jBPM².

Se utiliza Nuxeo como gestor documental, para almacenar los ficheros de los documentos, gestionando activos digitales como imágenes tomadas a los documentos. Nuxeo permite a los arquitectos y desarrolladores configurar y personalizar la aplicación para adaptarse al contexto del proyecto, mediante la posibilidad de crear, implementar y ejecutar aplicaciones basadas en el contenido. Además, posee un grupo de características que le permite su integración para el desarrollo del sistema.

1.4.2.5 Servidor de aplicaciones web

Apache Tomcat

Apache Tomcat (Jakarta Tomcat o simplemente Tomcat) funciona como un contenedor de servlets³ desarrollado bajo el proyecto Jakarta en la Apache Software Fundación. Implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. (26)

Tomcat es un servidor web con soporte de servlets y JSPs, no es un servidor de aplicaciones. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache. (26)

Puede funcionar como servidor web por sí mismo, en sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que fue escrito en java funciona en cualquier sistema operativo que disponga la máquina virtual Java (26).

El uso de Apache Tomcat dentro del desarrollo del subsistema, es de gran ayuda ya que al ser un contenedor web escrito en Java funciona en cualquier sistema operativo que disponga de una máquina virtual Java previendo posible migración de sistema operativo. El contenedor web Apache Tomcat se encuentra integrado al framework Grails, permitiendo un rápido despliegue de las aplicaciones web, en la actualidad se usa en numerosas aplicaciones web.

² jBPM es un código abierto motor de flujo de trabajo escrito en Java. jBPM administra las instancias de procesos, además de gestionar y conservar el estado de las instancias de proceso en todo momento. también proporciona diversas herramientas para crear, implementar, ejecutar y gestionar los procesos de negocio a lo largo de su ciclo de vida.

³Servlets: Son objetos que corren dentro y fuera del contexto. Refiere a pequeños programas que se ejecutan en el contexto de un navegador web.

1.4.3 Tecnologías de desarrollo

La tecnología es una característica en la aplicación de conocimientos científicos acumulados con el fin de aplicar los procedimientos técnicos necesarios que conduzcan a las soluciones óptimas en la capacidad de construir, a partir de una gran variedad de objetos y herramientas, el modo de emplear el desarrollo y perfección con vistas a modificar favorablemente el entorno de trabajo (27).

JQuery

Es una librería de JavaScript rápida y concisa que simplifica el trabajo con documentos HTML, accede a los objetos del Modelo de Objetos de Documento (DOM) de un modo simplificado. JQuery ha sido diseñado para cambiar la forma de escribir JavaScript. Utiliza un interesante concepto para hacer código corto y simple, tiene manejadores de eventos. Otro tema que JQuery resuelve con facilidad es el de los efectos, añade dinamismo visual a la presentación del sitio, como son añadirle funcionalidad, tanto al código como al resto de los elementos, lo que permite no preocuparse por la compatibilidad de navegadores (28).

La librería JQuery en resumen aporta las siguientes ventajas:

- Ahorro de muchas líneas de código.
- Transparencia en el soporte de la aplicación para los navegadores principales.
- Provee un conjunto de funciones para animar el contenido de la página en forma muy sencilla.

Se selecciona por su alto grado de aceptación por parte de los programadores y su amplia introducción en el mercado, es una de las librerías más potentes integradas a Grails, siendo muy utilizado y beneficioso en el uso del lenguaje de programación JavaScript, permitiendo mayor facilidad en el desarrollo del sistema. Es un producto serio, estable y documentado.

1.4.3.1 Framework

Grails 2.1.1

Es un framework⁴ para aplicaciones web, libre, desarrollado sobre el lenguaje de programación Groovy. Grails pretende ser un framework con un marco de trabajo altamente productivo siguiendo paradigmas

⁴Marco de Trabajo. [framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos]

tales como convención sobre configuración o no te repitas DRY⁵, proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de configuración al programador (29).

Dentro de los objetivos que persigue Grails para su desarrollo se encuentran:

- Ofrecer un framework web de alta productividad para la plataforma Java.
- Reutilizar tecnologías Java ya probadas como Hibernate y Spring bajo una interfaz simple y consistente.
- Ofrecer un framework consistente que reduzca la confusión y que sea fácil de aprender.
- Patrones de visualización potente y fácil de usar con GSP (Groovy Server Pages).
- Bibliotecas de etiquetas dinámicas para crear fácilmente componentes web.
- Proporciona un entorno completo de desarrollo, incluyendo un servidor web y recarga automática de recursos.

Dentro de las características principales de Grails están (30):

- Alta productividad: Grails posee tres características que intentan incrementar su productividad en comparación con los Framework Java tradicionales:
 - Inexistencia de configuración XML.
 - Entorno de desarrollo preparado para funcionar desde el primer momento.
 - Funcionalidad disponible mediante métodos dinámicos.
- Integración con la plataforma Java: Grails al estar construido y basado sobre la plataforma Java, ofrece una integración transparente con clases mapeada mediante el Framework Hibernate ORM⁶, por lo que aplicaciones existentes que utilicen Hibernate pueden utilizar Grails sin recompilar el código o reconfigurar las clases Hibernate, permitiendo que con las clases Java mapeadas puedan utilizar scaffolding⁷, además de que las capacidades de Grails están totalmente disponibles para estas clases y las aplicaciones que las usan.

⁵No te Repitas (Don't Repeat Yourself, por sus siglas en inglés): es una práctica de programación de computadoras de evitar la redundancia de código en las aplicaciones informáticas.

⁶Mapeo de Objeto Relacional (Object Relational Mapping por sus siglas en inglés).

⁷Técnica de programación soportada por algunos marcos de trabajo Modelo Vista Controlador, en la cual el programador escribe una especificación que describe como la base de datos de la aplicación debe ser usada. El compilador utiliza esta especificación para generar código que la aplicación puede usar para crear, leer, actualizar y eliminar entradas de la base de datos.

- Persistencia: El modelo de datos en Grails se almacena en la base de datos utilizando Grails Object Relational Mapping (GORM).

Bootstrap

Bootstrap es un framework que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Ha sido desarrollado por Twitter que recientemente liberó su versión 2.0. La mayor ventaja es que se puede crear interfaces que se adapten a los distintos navegadores, apoyándose en un framework potente con numerosos componentes webs que minimizarán el esfuerzo y tiempo en el desarrollo de las aplicaciones (31).

Características principales de Bootstrap:

- Bootstrap ofrece una serie de plantillas CSS y ficheros JavaScript que permite integrar el framework de forma sencilla y potente en proyectos webs.
- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tabletas electrónicas y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías JavaScript, por ejemplo JQuery.
- Ofrece estándares como CSS3/HTML5.
- Es un framework ligero que se integra de forma limpia en el proyecto actual.
- Dispone de distintos layout predefinidos con estructuras de distintos diseños fluidos.

Se selecciona Bootstrap porque es un conjunto de estilo CSS y lenguaje HTML que ofrece una solución flexible y de mucha más potencia que un CSS normal, facilitando el uso de animaciones, estilos y funciones de colores en la aplicación, ya que ofrece plantillas para maquetar, estilos para tipografías, formularios o diferentes botones de todo tipo, utilizando su librería; además propicia interfaces adaptables para todo tipo de navegador.

1.4.4 Lenguajes

1.4.4.1 Lenguajes de Modelado

El lenguaje de modelado es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software orientado a objetos. Se usan en combinación de una metodología de desarrollo de software.

Lenguaje Unificado de Modelado (UML)

El Lenguaje unificado de modelado (UML) por sus siglas en inglés Unified Modeling Language, prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, que permite especificar, visualizar y construir los artefactos de los sistemas de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios.

El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos (30).

En términos muy generales UML define una notación que se expresa con diagramas. Para mostrar las diferentes perspectivas del modelado, UML define 9 tipos de diagramas: (32)

- Diagramas de casos de uso
- Diagramas de clases
- Diagramas de componentes
- Diagramas de despliegue
- Diagramas de objetos
- Diagramas de colaboración
- Diagramas de estados y transiciones
- Diagramas de actividades

Se seleccionó como Lenguaje de Modelado a UML, por soportar la metodología de desarrollo RUP, es orientado a sistemas y no a procesos, está consolidado como lenguaje estándar de análisis y diseño. Los beneficios que se consiguen al utilizar UML son varios, por un lado el uso de lenguajes visuales facilitan su asimilación y entendimiento por parte del equipo de desarrollo; el tiempo invertido en el desarrollo de la arquitectura se minimiza, obteniendo como ventajas que la trazabilidad y la documentación del proyecto se realiza de una forma ordenada y guiada por casos de uso.

1.4.4.2 Lenguaje de Desarrollo

Groovy

Groovy es un lenguaje dinámico y ágil para la plataforma JAVA, se basa en los puntos fuertes de Java aunque tiene características inspirado en lenguajes como Python, Ruby y Smalltalk.

Dentro de las características más destacadas se encuentran: (33)

- Aumento la productividad de los desarrolladores de código mediante la reducción de los andamios en el desarrollo web, interfaz gráfica de usuario, bases de datos o aplicaciones de consola.
- Dinámico e interpretado, ya que al definir variables, no es necesario declarar el tipo de dato de las mismas, no tiene una sintaxis estricta en el cuerpo principal de los scripts y todo lo que se escribe en Groovy es ejecutado por el intérprete.
- Se integra perfectamente con todas las clases existentes de Java y las bibliotecas.

Java

Java es un lenguaje de desarrollo de propósito general, válido para realizar todo tipo de aplicaciones profesionales, fue creada por la empresa Sun Microsystems en 1995. Java con respecto a otros lenguajes de programación como por ejemplo C++ posee un modelo de objetos mucho más simple y elimina herramientas de bajo nivel (34).

Una de las cualidades importantes de Java es la independencia de la plataforma e independencia de arquitectura, lo cual permite que programas escritos en el lenguaje Java puedan ser ejecutados en cualquier tipo de hardware o dispositivos siempre que se cuente con una máquina virtual.

El uso del lenguaje de programación Java le permite al usuario realizar varias actividades entre las que se encuentran:

- Desarrollar software en una plataforma y ejecutarlo prácticamente en cualquier otra plataforma.
- Crear programas para que funcionen en un navegador web y en servicios web.
- Combinar aplicaciones o servicios que usan el lenguaje Java para crear servicios o aplicaciones totalmente personalizados.
- Compatibilidad con todo el código heredado Java.
- La posibilidad de usar Groovy para aumentar la productividad de los desarrolladores en áreas como:

- ✓ Código de integración entre componentes.
- ✓ Código de configuración.
- ✓ Lenguajes de dominio específico.
- ✓ Capa de presentación.
- ✓ Automatización de tareas y desarrollo de Scripts.
- ✓ Programación dinámica (creación y alteración de tipos de dato en tiempo de ejecución).

La combinación de los lenguajes Groovy y Java en el desarrollo del sistema informático Xabal Arkheia es aprovechada por los desarrolladores y por el equipo de arquitectura del mismo ya que constituye la plataforma de nueva generación en el desarrollo de aplicaciones Java, al aportar lo mejor de los lenguajes estáticos y dinámicos.

JavaScript

JavaScript es un lenguaje de programación utilizado para unir un conjunto de tecnologías que permite a los desarrolladores crear acciones en las páginas web. El lenguaje de programación está basado en objetos utilizados para acceder a objetos en aplicaciones, principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas, no requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Modelo de Objetos de Documento (DOM), donde su principal importancia radica en que anteriormente se venía utilizando en páginas web HTML para realizar operaciones y en el marco de la aplicación cliente, sin acceso a funciones del servidor, en el caso de JavaScript se ejecuta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML (35).

Dentro de los medios que proporciona el lenguaje de programación JavaScript se encuentran:

- Controlar las ventanas del navegador y el contenido que muestran.
- Programar páginas dinámicas simples sin tener que matar moscas a cañonazos de Java.
- Capturar los eventos generados por el usuario y responder a ellos sin salir a Internet.
- Comprobar los datos que el usuario introduce en un formulario antes de enviarlos.
- Comunicarse con el usuario mediante diversos métodos.

Este lenguaje es seleccionado por su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Permite crear efectos especiales en las páginas y definir interactividades con el usuario.

HTML

Lenguaje de Marcado de Hipertexto (HTML) conocido así por sus siglas en inglés (Hyper Text Markup Lenguaje), es un lenguaje de marca orientado a la publicación de documentos en Internet o sea es el lenguaje con que se escriben las páginas web. Está constituido por elementos que el navegador interpreta y las despliega en la pantalla de acuerdo a su objetivo. Los documentos HTML están formados por una serie de bloques de texto con una entidad lógica (titulares, párrafos, listas, entre otros). La interpretación de estas entidades se deja al navegador, lo cual da una gran flexibilidad a la presentación del documento que se presente, ejemplo en terminales gráficos o de texto.

Este lenguaje se utilizará como parte de la codificación de las páginas servidoras y como resultado del procesamiento de las mismas, siendo de codificación por detrás de las vistas que ve el cliente.

1.5 Conclusiones

En el presente capítulo se realizó un estudio al proceso archivístico para un mejor entendimiento del problema planteado, lo que ha permitido analizar el estado actual de la archivística en lo referente al proceso de almacenamiento y conservación de archivo. Para ello, se estudiaron conceptos y se analizaron soluciones informáticas en relación con el almacenamiento y conservación para los procesos archivísticos.

Partiendo de estudios y análisis realizados para el desarrollo del sistema en el marco de la definición de la metodología, herramientas, tecnologías a utilizar en el desarrollo del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia, permitió seleccionar a *RUP (Proceso Unificado de Racional)* como metodología de desarrollo. La utilización del *Visual Paradigm For UML 8.0* como herramienta de modelado visual para la realización de los diagramas UML, mientras que para el desarrollo del sistema, se contará con el entorno integrado multiplataforma *SpringSource Tool Suite (STS) 2.5*, el *TortoiseSVN 1.7* como cliente para administrar archivos y repositorios, el gestor de base de datos *PostgreSQL 9.1* y el gestor de documentos *Nuxeo*, empleando al contenedor web *Apache Tomcat* como servidor de aplicaciones web. Los framework para aplicaciones web *Grails 1.3.7* y *Bootstrap*, y la librería de JavaScript *JQuery* conformarán las tecnologías a utilizar. Además, se utilizará como lenguaje de

modelado a *UML (Lenguaje Unificado de Modelado)*, mientras que los lenguajes de desarrollo estarán compuestos por *Groovy* y *Java* como lenguajes de programación del lado del servidor, *HTML* y *JavaScript* como lenguajes de programación del lado del cliente.

Capítulo.2 Descripción de la solución propuesta

2.1 Introducción

El presente capítulo plantea la propuesta de solución del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia. Cuenta con los procesos de negocio, los diagramas del modelo de dominio, requisitos funcionales y no funcionales, los diagramas y la especificación de los casos de uso de los módulos del subsistema.

2.2 Propuesta de solución

Después de realizar un estudio de la situación problemática antes planteada y teniendo como presente el uso del ambiente de trabajo seleccionado para el desarrollo del sistema Xabal Arkheia, se propone como solución la implementación de los módulos Almacenamiento y Conservación de archivo.

Mediante la digitalización de los procesos de almacenamiento y conservación de archivos, se logrará un mayor control de la gestión, organización, tratamientos y procesos que se realicen, lo que ayudará a preservar el estado físico de la documentación.

El proceso de almacenamiento se encarga de almacenar y gestionar las estructuras, medios y unidades de conservación que defina la institución. Ayuda al proceso de gestión de ubicación física de los documentos, simplifica la búsqueda de información para el manejo, control y gestión de estos, minimizando la ocurrencia de riesgo como la pérdida de documentos en la institución. Para ello se cuenta con una estructura jerárquica previamente definida, que permite al usuario encargado crear y/o explorar la estructura física de la institución permitiendo realizar la ubicación física de las unidades de conservación, de las cuales se obtiene una breve descripción y se le puede realizar un conjunto de acciones; así como crear niveles estructurales para almacenar las unidades de conservación rigiéndose por las reglas de jerarquía definidas previamente.

El proceso de conservación es el encargado de gestionar las acciones dentro del área de conservación, que incluye la ficha de conservación y los servicios de restauración y encuadernación. Gestiona todo un conjunto de datos mediante el uso de la ficha de conservación que permite conocer el estado en que se encuentra el documento al que se le está realizando el proceso, dada la información que se genera, se realiza un seguimiento y control a los problemas encontrados (enfermedades, plagas y condiciones ambientales) que afectan la conservación de los documentos. Además, posibilita la aplicación de los servicios de restauración y encuadernación completando el proceso de conservación.

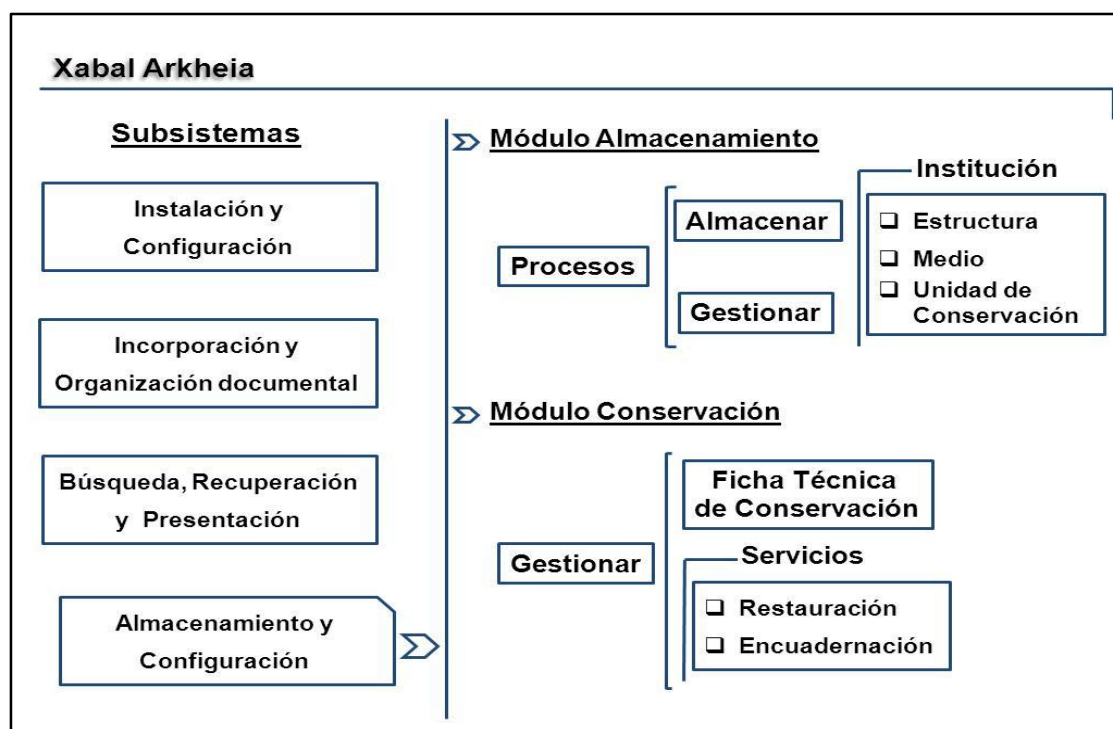


Figura.2 Propuesta de solución

2.3 Proceso de negocio del módulo Almacenamiento

Descripción del flujo básico

Para el proceso de almacenamiento es necesidad indispensable que la institución haga entrega al coordinador de los medios y unidades de conservación pendientes a ubicar. El coordinador ubica los medios y unidades de conservación que le han sido entregados a partir de la estructura predefinida, y del objetivo de estos (respondiendo a qué se depositará en ellos).

Descripción de los flujos paralelos

El coordinador define cuál será la estructura por la cual se guiará para el almacenamiento de los medios y/o unidades de conservación que le haga entrega la institución, para ello debe tener en cuenta cuál es la política a seguir y para qué será utilizado. Si los locales de la institución ya están completos el coordinador debe realizar una solicitud a los directivos para que le asignen un nuevo local de almacenamiento.

Diagrama de proceso Almacenamiento

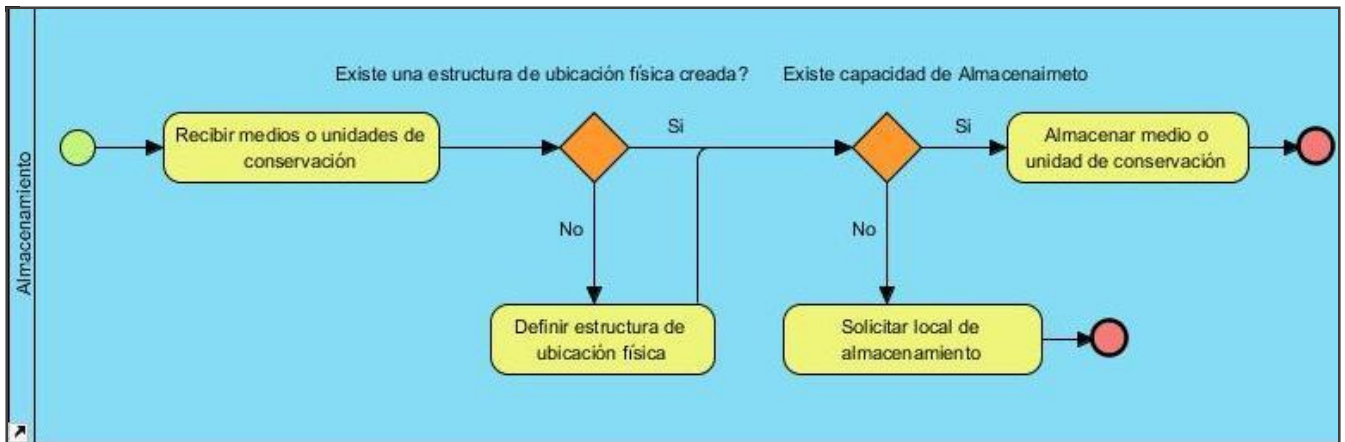


Figura.3 Diagrama de proceso del módulo Almacenamiento.

2.4 Proceso de negocio del módulo Conservación

El módulo de Conservación está compuesto por dos procesos de negocio, el de conservar documentos y el de realizar servicio de restauración y/o encuadernación.

2.4.1 Proceso de negocio Conservar Documento

Descripción del flujo Básico

El especialista de conservación realiza un diagnóstico técnico de los documentos de la oficina para determinar su estado de conservación. El especialista le aplica los tratamientos a los documentos que así lo requieran, luego del diagnóstico técnico. Se obtienen y registran los resultados de la aplicación de los tratamientos realizados a los documentos.

Diagrama de Proceso Conservar Documentos.

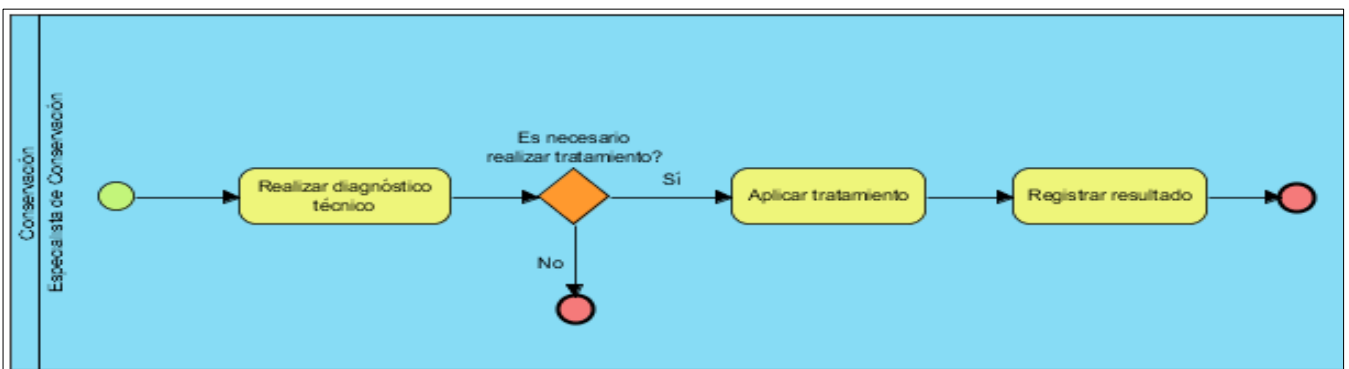


Figura.4 Diagrama de proceso de negocio Conservar Documento.

2.4.2 Proceso de negocio de módulo Conservación para realizar servicio de restauración y/o encuadernación

Descripción del flujo básico

El especialista de conservación recibe las solicitudes de restauración y/o encuadernación de piezas que traen los clientes, realiza el trabajo de restauración y/o encuadernación de las mismas y luego se entrega.

Descripción de los flujos Paralelos

Si no es posible procesar la solicitud, el coordinador le informa al cliente que no puede ser atendido su pedido.

Diagrama de proceso para realizar servicio de restauración y/o encuadernación.

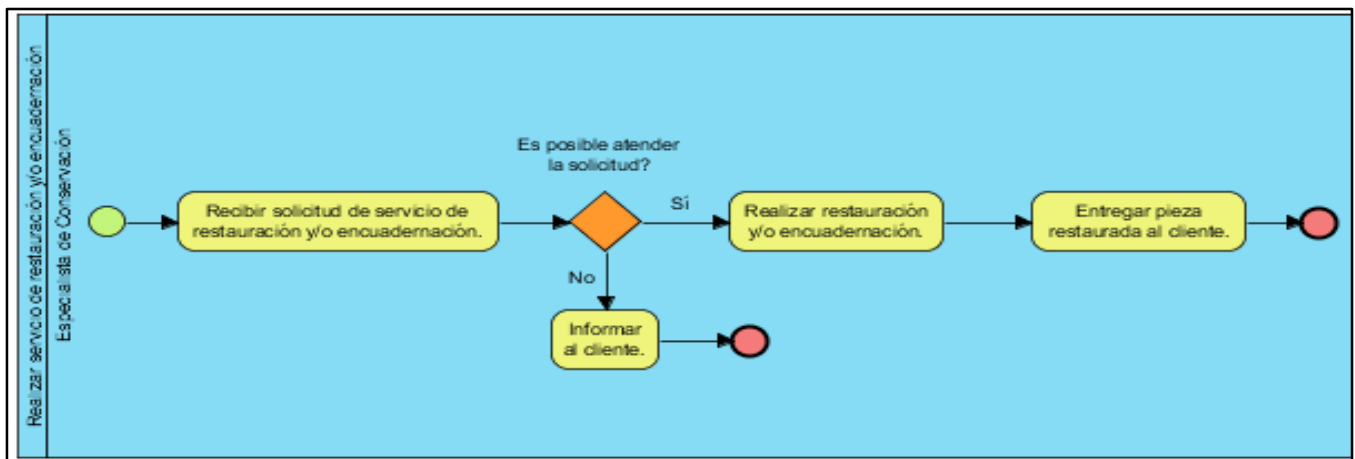


Figura.5 Diagrama de proceso de negocio para realizar servicio de restauración y/o encuadernación.

2.5 Modelo de dominio

El modelo de dominio permite observar de manera visual los principales conceptos que se utilizan en el dominio del subsistema. Además brinda la posibilidad de entender el problema que el subsistema resuelve en relación a su contexto.

A continuación se muestra el diagrama de clase del modelo de dominio del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia.

Modelo de dominio del subsistema Almacenamiento y Conservación.

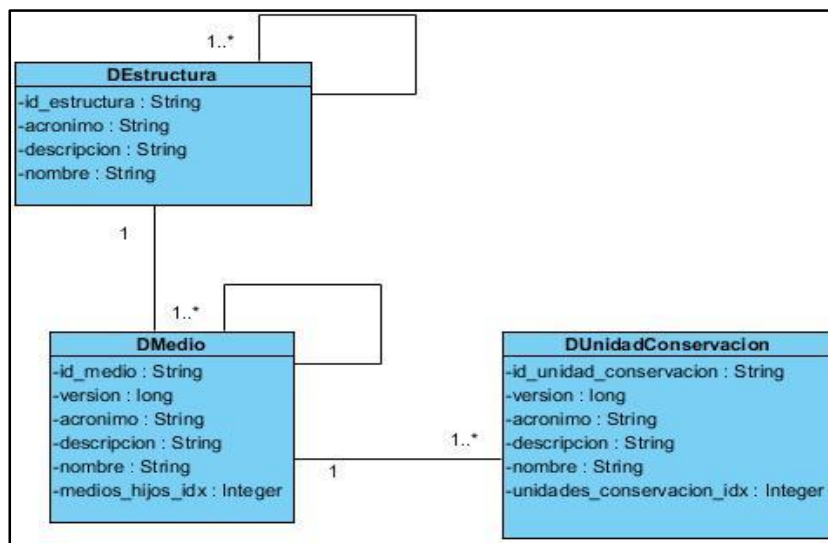


Figura.6 Modelo de dominio del módulo Almacenamiento.

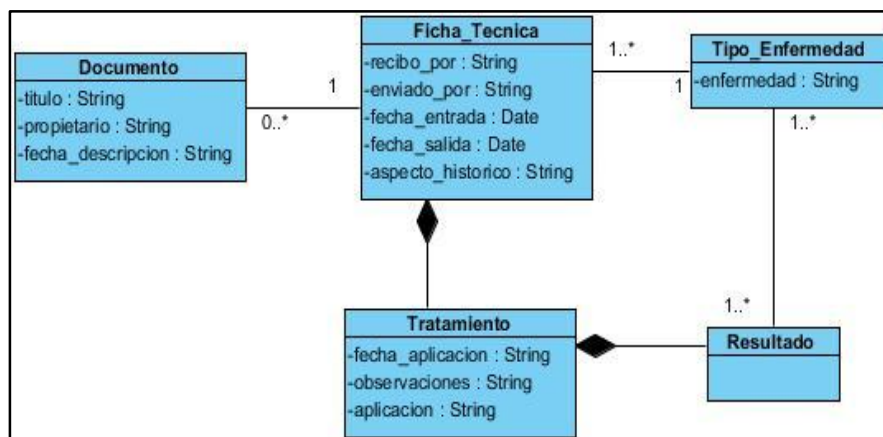


Figura.7 Modelo de dominio para Conservar Documento del módulo de Conservación.

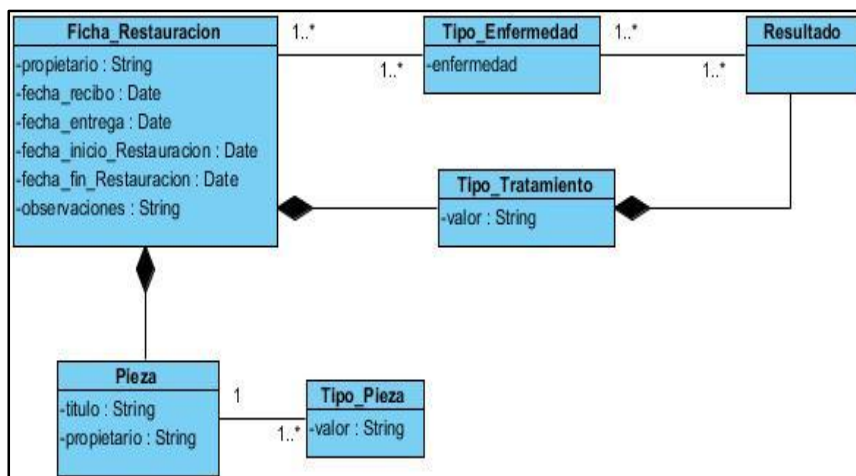


Figura.8 Modelo de Dominio para para realizar servicio de restauración y/o encuadernación del módulo Conservación

2.6 Requisitos funcionales

Un requisito funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. Son complementados por los requisitos no funcionales, que se enfocan en cambio en el diseño o la implementación (30).

Típicamente, un analista de requisitos genera requisitos funcionales luego de diagramar los casos de uso. Sin embargo, esto puede tener excepciones, ya que el desarrollo de software es un proceso iterativo y algunos requisitos son previos al diseño de los casos de uso. Ambos elementos (casos de uso y requisitos) se complementan en un proceso bidireccional.

2.6.1 Requisitos funcionales del módulo Almacenamiento

RF1-Adicionar los datos de un nodo de la estructura de ubicación física.

RF2-Modificar los datos de un nodo de la estructura de ubicación física.

RF3-Cambiar la ubicación física de un nodo de la estructura de ubicación física.

RF4-Visualizar los datos de detalle de un nodo de la estructura de ubicación física.

RF5-Eliminar un nodo de la estructura de ubicación física.

RF6-Explorar la estructura de ubicación física.

RF7-Adicionar una categoría a la estructura de ubicación física.

RF8-Modificar una categoría de la estructura de ubicación física.

RF9-Eliminar una categoría de la estructura de ubicación física.

RF10-Registrar los datos de una unidad de conservación de la estructura de ubicación física.

RF11-Ubicar una unidad de conservación en la estructura de ubicación física.

RF12-Modificar los datos de una unidad de conservación de la estructura de ubicación física.

RF13-Cambiar la ubicación física de una unidad de conservación de la estructura de ubicación física.

RF14-Cambiar el estado de una unidad de conservación.

RF15-Visualizar los datos de detalle de una unidad de conservación de la estructura de ubicación física.

RF16-Eliminar una unidad de conservación de la estructura de ubicación física.

RF17-Buscar las unidades de conservación registradas a partir de los criterios de búsquedas introducidos.

RF18-Visualizar la ubicación física de una unidad de conservación.

2.6.2 Requisitos funcionales del módulo Conservación

RF1-Registrar los datos de un diagnóstico técnico realizado.

RF2-Modificar los datos de un diagnóstico técnico realizado.

RF3-Visualizar los datos de detalle de un diagnóstico técnico realizado.

RF4-Buscar los diagnósticos técnicos registrados a partir de los criterios de búsquedas introducidos.

RF5-Asociar los datos de un tratamiento a su diagnóstico técnico correspondiente.

RF6-Modificar los datos de un tratamiento.

RF7-Visualizar los datos de detalle de un tratamiento.

RF8-Adicionar el resultado de un tratamiento.

RF9-Modificar los datos de un resultado.

RF11-Visualizar los datos de detalle de un resultado.

RF12Registrar los datos de una ficha de restauración y encuadernación.

RF13-Modificar los datos de una ficha de restauración y encuadernación.

RF14-Visualizar los datos de detalle de una ficha de restauración y encuadernación.

RF15-Buscar las ficha de restauración y encuadernación registradas a partir de los criterios de búsquedas introducidos.

2.7 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Son características que hacen al producto atractivo, usable, o confiable. Son fundamentales para el éxito del mismo.

Requerimientos de Hardware:

PC Cliente:

- RAM: 512 MB.
- Tarjeta de Red.

PC Servidor:

- RAM: 2GB para servidor de Base de Datos.
- RAM: 1GB para servidor de Aplicaciones.
- Disco Duro: 15 GB para base de datos y aplicaciones.
- Disco Duro: 20 GB para base de datos de Nuxeo.
- Tarjeta de Red.

Requerimientos de Software:

PC Cliente:

- Navegador Web: Navegador Web Mozilla Firefox 17 o superior, Google Chrome 23 o superior.
- Lector de pdf: Adobe Reader 10.

PC Servidor:

- Servidor de aplicaciones web: Apache Tomcat 7.0.
- Servidor de base de datos: PostgreSQL 9.1.
- Máquina Virtual de Java 7.
- Sistema Operativo: Windows o Linux.

Requerimientos de apariencia o interfaz externa:

- El sistema sigue los principios de diseño establecidos para la línea de productos XABAL referente a sistemas de administración pública realizados en la universidad.
- El sistema debe tener una apariencia profesional y un diseño gráfico sencillo, con la utilización de las tonalidades de los colores rojo, blanco y gris fundamentalmente.
- El texto incluido en las páginas de la aplicación debe ser de tipo arial con un tamaño 11.

Restricciones en el diseño y la implementación:

- Framework Grails 2.1.1.
- Framework Bootstrap.

Requerimientos de Seguridad:

Confidencialidad: La información controlada por el sistema debe estar protegida de acceso no autorizado por lo cual se establecerá un nivel de acceso a la aplicación mediante la gestión de roles.

Requerimientos de Rendimiento:

El tiempo promedio de respuesta por transacción es de no más de 30 segundos, debido a que el sistema es una herramienta web que realiza transferencias de documentos. Este tiempo se ve en gran medida condicionado por el tamaño del archivo con el que se esté trabajando.

Requerimientos de Portabilidad:

En el servidor, el sistema podrá ser usado sin importar el sistema operativo, siempre que en el mismo pueda ejecutarse la máquina virtual de java.

2.8 Actores del subsistema

Un actor es una idealización de una persona externa, de un proceso que interactúa con un sistema, un subsistema, o una clase. Un actor caracteriza las interacciones que los usuarios exteriores pueden tener con el sistema. En tiempo de ejecución, un usuario físico puede estar limitado a los actores múltiples dentro del sistema. Diferentes usuarios pueden estar ligados al mismo actor y por lo tanto pueden representar casos múltiples de la misma definición de actor (30).

Los actores del sistema son los encargados de automatizar las actividades del subsistema e interactuar con el mismo.

Actores del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia.

Actor del subsistema	Descripción
Usuario	Encargado de la autenticación de las personas que interactúan con el sistema.
Coordinador de Almacenamiento	Es el encargado de adicionar, modificar, eliminar, cambiar, explorar un nodo de la estructura de ubicación física. También es el responsable de adicionar una unidad de conservación. Y modificarla, eliminarla, moverla y cambiar su estado.
Especialista de Conservación	Se encarga de realizar todas las acciones referentes al módulo Conservación como son: registrar, modificar, o ver detalles de una ficha técnica de conservación, de una ficha de restauración y encuadernación, un tratamiento y un resultado.

2.9 Diagrama de casos de uso

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

Diagrama de casos de uso del subsistema Almacenamiento y Conservación.

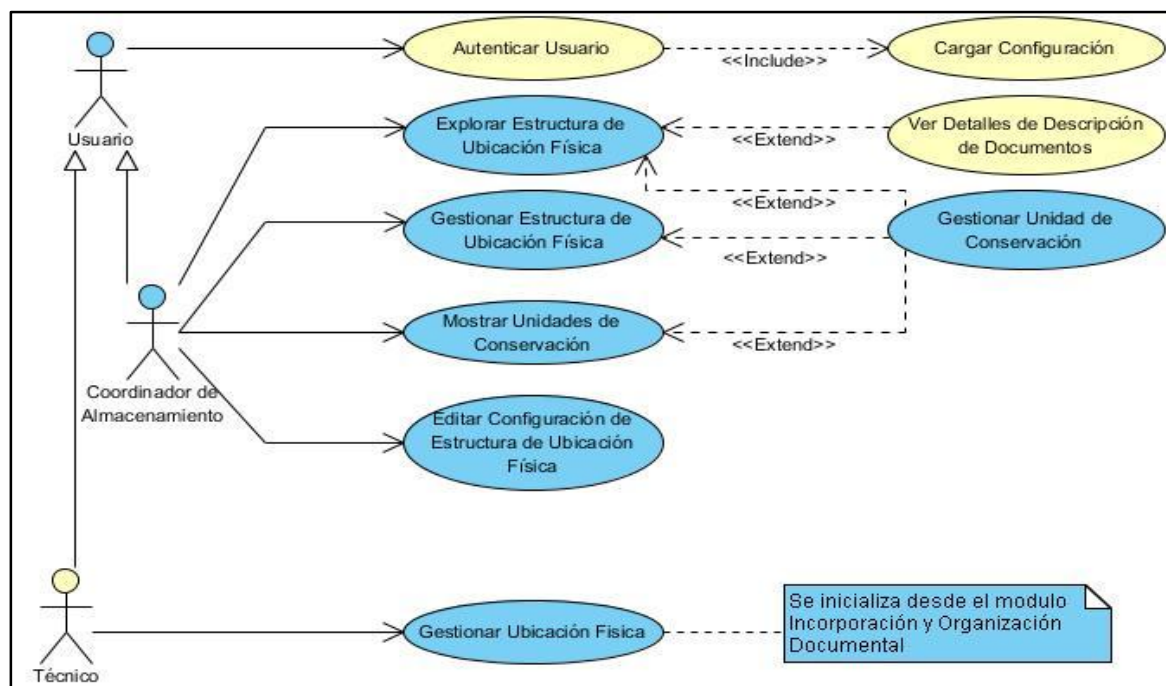


Figura.9 Diagrama de casos de uso del módulo Almacenamiento.

En el diagrama se representa los Casos de Uso Autenticar Usuario, Cargar Configuración y Ver Detalles de Descripción de Documento, debido a que estas funcionalidades son utilizadas o las realiza el módulo pero no son implementadas en él. Además de que cuenta con un actor externo (Técnico) quien utiliza funcionalidades dentro del módulo.

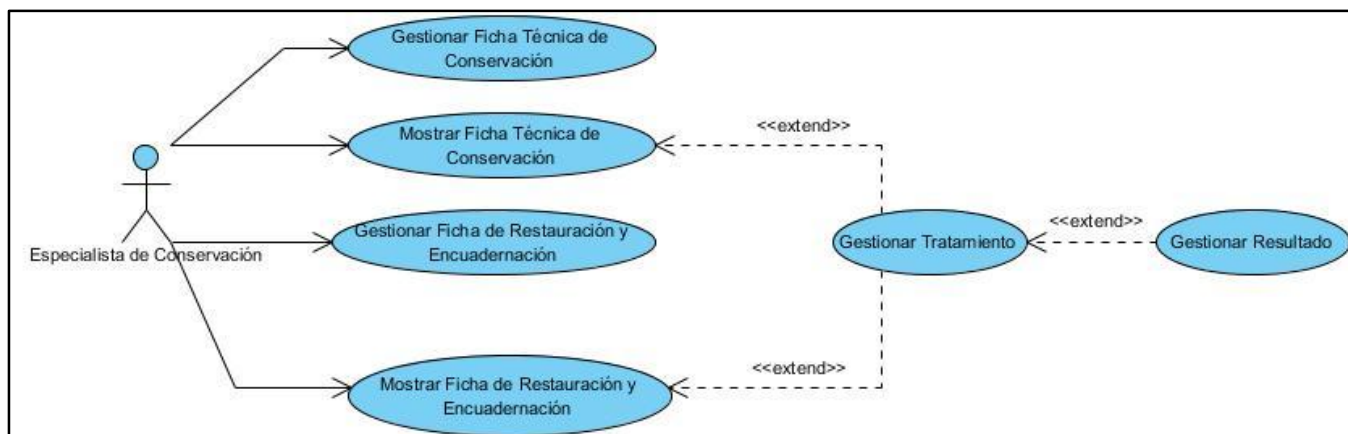


Figura.10 Diagrama de casos de uso del módulo Conservación.

2.10 Casos de uso

Los casos de Uso son artefactos narrativos que describen el comportamiento desde el punto de vista del usuario.

A continuación se describen brevemente los casos de uso de los módulos Almacenamiento y Conservación.

2.10.1 Casos de uso del módulo Almacenamiento

CU 1. Gestionar Estructura de Ubicación Física.

Objetivo	Adiciona, modificar, eliminar y/o cambiar la ubicación de un nodo de la estructura de ubicación física donde se almacenan los documentos.
Actores	Coordinador de Almacenamiento (Inicia): Adiciona, modifica, elimina y/o cambia la ubicación de un nodo de la estructura de ubicación física.
Resumen	El actor solicita realizar una acción sobre la estructura de ubicación física, el sistema muestra el formulario que permite realizar la acción solicitada. El actor indica las operaciones a realizar por el sistema resultando la estructura de ubicación física almacenada en la BD.

CU 2. Explorar Estructura de Ubicación Física.

Objetivo	Explorar la estructura de ubicación física.
Actores	Coordinador de Almacenamiento (Inicia): explora la estructura de ubicación física.
Resumen	El actor solicita explorar la estructura de ubicación física, el sistema permite navegar por la estructura de ubicación física creada previamente y muestra las acciones que pueden realizarse desde la misma.

CU 3. Editar Configuración de Estructura de Ubicación Física.

Objetivo	Editar la configuración de la Estructura de Ubicación Física.
Actores	Coordinador de Almacenamiento: (Inicia) Edita las estructuras de ubicación física ya definidas, adiciona, modifica, o elimina una categoría y sus datos correspondientes de la configuración de la estructura de ubicación física.
Resumen	El actor selecciona realizar una acción sobre la configuración de la estructura de ubicación física, el sistema permite adicionar, modificar o eliminar una categoría de la estructura de ubicación física configurada previamente.

CU 4. Gestionar Unidad de Conservación.

Objetivo	Ubicar, modificar, mover, eliminar, cambiar el estado, o ver la unidad de conservación.
Actores	Coordinador de almacenamiento (Inicia): Ubica, modifica, elimina, mueve y cambia el estado de una unidad de conservación.
Resumen	El caso de uso inicia cuando el actor decide realizar las acciones permitidas sobre una unidad de conservación, el sistema permite realizar la acción solicitada por el actor y termina el caso de uso.

CU 5. Mostrar Unidades de Conservación.

Objetivo	Mostrar un listado de las unidades de conservación según los criterios de consulta.
Actores	Coordinador de Almacenamiento (Inicia)
Resumen	El actor selecciona la opción que permite consultar las unidades de conservación, el sistema muestra un formulario con los criterios de búsqueda que el actor debe introducir para obtener el listado de las unidades de conservación que cumplan con los criterios de búsqueda especificados.

CU 6. Gestionar Ubicación Física.

Objetivo	Modificar y mostrar la ubicación física de un documento dentro de la estructura de ubicación física.
Actores	Técnico.
Resumen	El técnico solicita realizar una acción sobre la ubicación física del documento, el sistema muestra el formulario que permite realizar la acción solicitada. El coordinador indica las operaciones necesarias a realizar por el sistema.

2.10.2 Casos de uso del módulo Conservación

CU 1. Gestionar Ficha Técnica de Conservación.

Objetivo	Registrar, modificar, o ver detalles de una ficha técnica de conservación
Actores	Especialista de Conservación: (Inicia) Registra, modifica, o ve los detalles de la ficha técnica de conservación.
Resumen	El caso de uso se inicia cuando el actor desea realizar alguna acción sobre la ficha técnica de conservación. El sistema permite realizar la acción solicitada por el actor y termina el caso de uso.

CU 2. Gestionar Tratamiento.

Objetivo	Registrar, modificar, o ver detalles de un tratamiento.
Actores	Especialista de Conservación: (Inicia) Registra, modifica o ve los detalles del tratamiento.
Resumen	El caso de uso se inicia cuando el actor desea realizar alguna acción sobre el tratamiento. El sistema permite realizar la acción solicitada por el actor y termina el caso de uso.

CU 3. Gestionar Resultado.

Objetivo	Registrar, modificar o ver detalles de un resultado.
Actores	Especialista de Conservación: (Inicia) Registra, modifica o ve los detalles del resultado.
Resumen	El caso de uso se inicia cuando el actor desea realizar alguna acción sobre el resultado. El sistema permite realizar la acción solicitada por el actor y termina el caso de uso.

CU 4. Mostrar Fichas Técnicas de Conservación.

Objetivo	Mostrar un listado de fichas técnicas de conservación según los criterios de consulta.
Actores	Especialista de Conservación (Inicia): Busca fichas técnicas de conservación.
Resumen	El actor selecciona la opción que permite consultar las fichas técnicas de conservación, el sistema muestra un formulario con los criterios de búsqueda que el actor debe introducir para obtener el listado de las fichas técnicas de conservación que cumplan con los criterios de búsqueda especificados.

CU 5. Gestionar Fichas de Restauración y Encuadernación.

Objetivo	Registrar, modificar, o ver Ficha de Restauración y Encuadernación.
Actores	Especialista de Conservación:(Inicia) Registra, modifica, y visualiza la Ficha de restauración y encuadernación.
Resumen	El caso de uso inicia cuando el actor decide realizar las acciones permitidas sobre una Ficha de Restauración y Encuadernación, el sistema permite realizar la acción solicitada por el actor y termina el caso de uso.

CU 6. Mostrar Fichas de Restauración y Encuadernación.

Objetivo	Mostrar un listado de fichas de restauración y encuadernación según los criterios de consulta.
Actores	Especialista de Conservación (Inicia): Busca fichas de restauración y encuadernación.
Resumen	El actor selecciona la opción que permite consultar las fichas de restauración y encuadernación, el sistema muestra un formulario con los criterios de búsqueda que el actor debe introducir para obtener el listado de las fichas de restauración y encuadernación que cumplan con los criterios de búsqueda especificados.

Mostrar: muestra una lista de unidades de conservación, ficha técnica de conservación o de restauración y encuadernación a partir de criterios de búsqueda introducidos, permitiendo realizar diferentes acciones sobre cada uno de estos objetos de la lista.

Ver detalles: se selecciona la opción “ver detalles” de una estructura, medio, unidad de conservación, ficha técnica de conservación o de restauración y encuadernación, el cual muestra los datos de estos en detalles.

2.11 Conclusiones

En presente capítulo se mostró una propuesta de solución después de realizar un estudio al problema planteado, permitiendo la digitalización y con esta un grupo de mejoras en el proceso de almacenamiento y conservación de archivos en una institución. Se realizó la descripción del proceso de negocio, además se exponen los diagramas de modelo de dominio de los módulos del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia. Se mencionó los requisitos funcionales y no funcionales que debe cumplir el subsistema. Para lograr un mayor entendimiento del negocio, se especificó los actores que interactúan con el subsistema y las actividades que estos realizan. Se mostró los diagramas de casos de usos de los módulos Almacenamiento y Conservación y un resumen de las descripciones de los casos de usos perteneciente a cada uno de ellos.

Capítulo.3 Diseño del subsistema

3.1 Introducción

En el capítulo se realiza el diseño del subsistema a partir de las funcionalidades antes mencionadas. Se realiza el diagrama de colaboración, el cual representa las interacciones entre los objetos de las diferentes clases. Se propone el patrón de arquitectura y los de diseño para el desarrollo del subsistema. Contiene el diagrama de paquetes, donde se organizan los elementos que subdividen el modelo del sistema, los diagramas de clases del diseño y de clases persistentes del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia.

En este capítulo se trabaja solamente con un caso de uso de cada módulo, haciendo menos extenso y engorroso el presente trabajo. Los demás casos de uso los podrá encontrar en la documentación que presenta el repositorio del proyecto Sistema de Gestión Documental para la oficina del Conservador de Trinidad y el Valle de los Ingenios.

3.2 Modelo de Diseño

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en otras palabras, cómo cumple el subsistema sus objetivos. Es muy importante, pues permite que se reduzcan los riesgos de cometer errores en la implementación, de construir un sistema inestable, sensible a los cambios o difícil de comprobar si es realmente funcional (30). Un buen diseño garantiza que el subsistema sea implementado sin ambigüedades.

3.2.1 Diagrama de interacción

Los diagramas de interacción muestran una interacción concreta entre un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos para cumplir ciertas tareas. Modelan el comportamiento dinámico del sistema, en otras palabras, el flujo de control en una operación (30).

Existen dos tipos de diagramas de interacción en UML (30):

- Diagramas de Colaboración (*dimensión estructural*).
- Diagramas de Secuencia (*dimensión temporal*).

Los diagramas de colaboración muestran las interacciones entre los objetos que componen el diagrama de clases del análisis, creando enlaces entre ellos y añadiendo mensajes a estos enlaces.

A continuación se muestra el diagrama de colaboración del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia, porque dado el uso de la herramienta de modelado Visual

Paradigm en el desarrollo del subsistema, permite generar los diagramas de secuencia después de conformado los diagramas colaboración.

Diagrama de colaboración del subsistema Almacenamiento y Conservación.

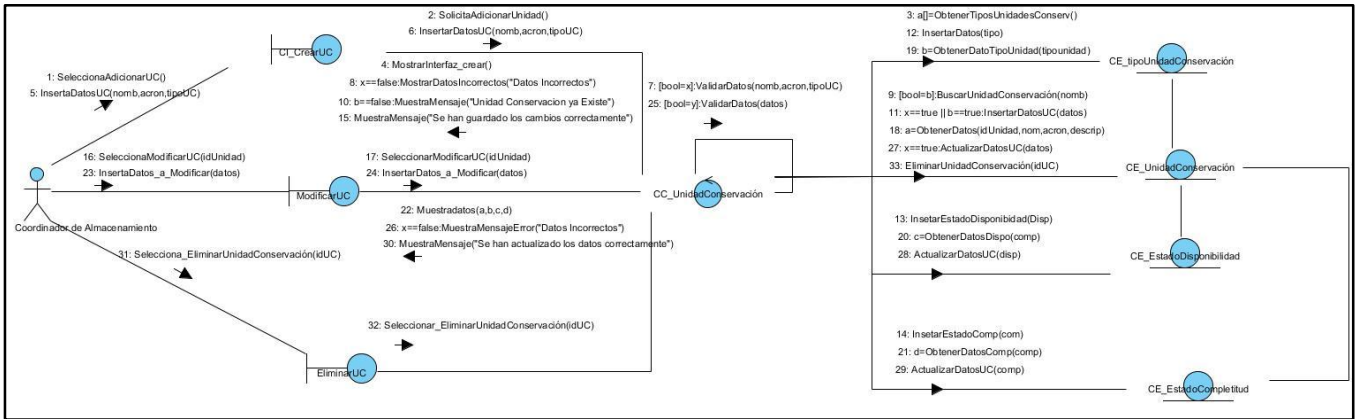


Figura.11 CU Gestionar Unidad de Conservación del módulo Almacenamiento. Secciones: Registrar, Modificar y eliminar unidad de conservación.

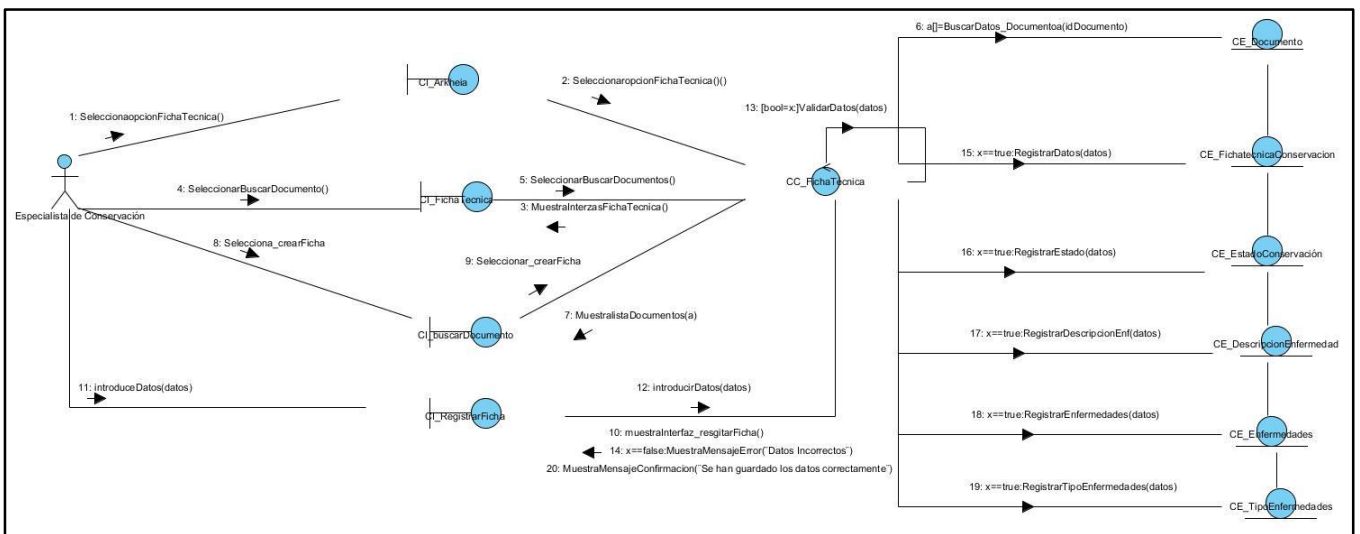


Figura.12 CU Gestionar Fichas Técnicas de Conservación del módulo Conservación. Sección: Registrar Ficha Técnica.

3.2.2 Diagrama de paquetes del subsistema

El diagrama de paquete es un mecanismo de organización de los paquetes y sus elementos que subdividen el modelo en otros más pequeños que colaboran entre sí (30).

Están constituidos por dos tipos de elementos:

- *Paquetes:* Permiten dividir un modelo en partes manejables mediante la agrupación de elementos que pueden ser casos de uso, clases o componentes. Pueden anidar otros paquetes dentro de sí.
- *Dependencias:* Indican que un elemento de un paquete requiere a otro de otro paquete distinto.

Este particionamiento debe hacerse sobre la base de los requerimientos funcionales y el dominio del problema; y debe ser reconocible por las personas con conocimiento del dominio.

A continuación se muestra del diagrama de paquetes del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia.

Diagrama de paquete del subsistema Almacenamiento y Conservación.

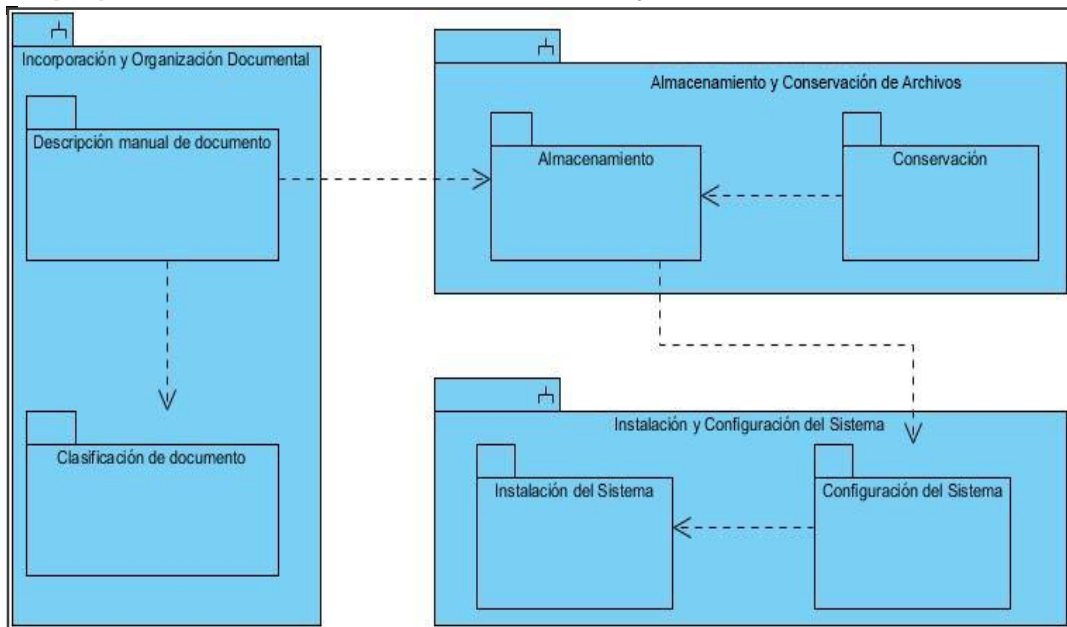


Figura.13 Diagrama de paquete del subsistema Almacenamiento y Conservación.

3.2.3 Patrón de arquitectura

Los patrones de arquitectura, son patrones de diseño que son el esqueleto de las soluciones a problemas comunes en el desarrollo de software, brindan una solución ya probada y documentada a problemas de desarrollo de software. Cada patrón describe un problema que ocurre una y otra vez en un ambiente, y luego describe el núcleo de la solución a ese problema.

Modelo Vista Controlador (MVC): Grails está basado en el patrón de arquitectura MVC. Este es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos (29).

Modelo, o capa de datos: Es el Sistema de Gestión de Base de Datos y gestiona la lógica de negocio. En ella se modelan las entidades del mundo real y se realizan las tareas necesarias para su manipulación, tal como su creación, almacenamiento y actualización (36). En Grails esta capa está compuesta por, capa de servicio, las clases de Dominio y el acceso a la base de datos.

Capa de Servicio: Controla el acceso de cualquier programa a la lógica de negocios, facilitando el mantenimiento y promoviendo la reutilización de código (36).

Clases de Dominio: Constituyen los datos persistentes y son entidades de la base de datos.

Controlador, o capa de control: Es el responsable de recibir los eventos de entrada desde la vista. En Grails, la capa de control es la encargada de lograr la interactividad entre las clases de dominio (o los servicios) y la capa de presentación a través de clases controladoras (36).

Vista, o Capa de presentación: La vista es la página HTML y el código que provee de datos dinámicos a esta. La capa de presentación en Grails viene representada por la tecnología Groovy Server Pages o GSP (36).

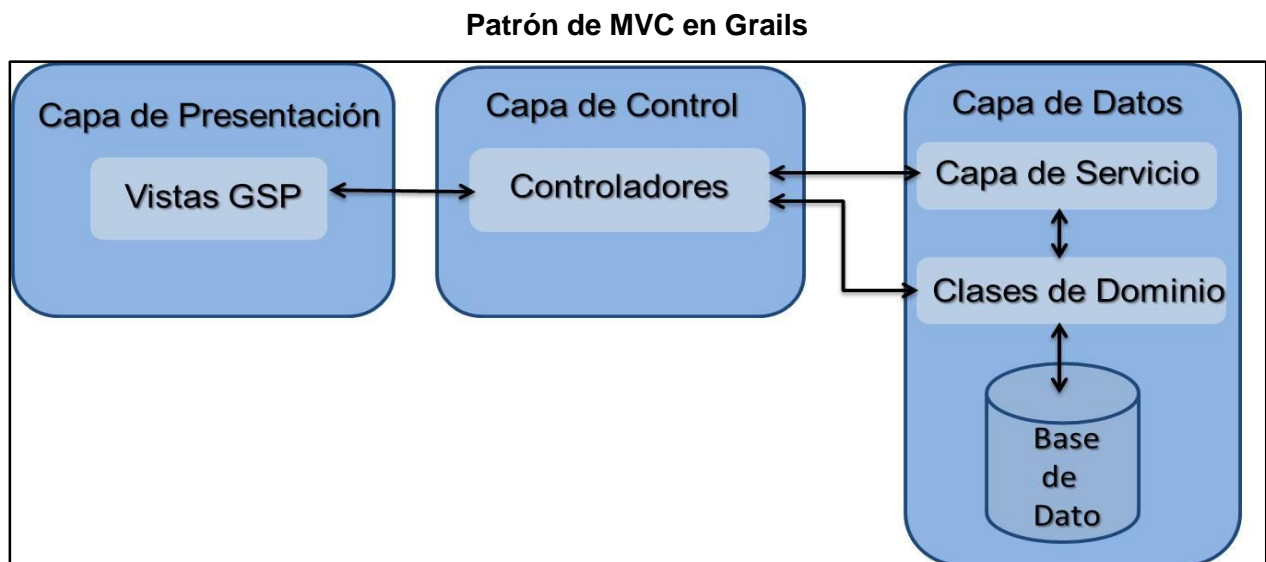


Figura.14 Patrón MVC en Grails.

3.2.4 Patrones de diseño GRASP

Patrón Experto

Es el encargado de asignar una responsabilidad al experto en información, es decir la clase que cuenta con la información necesaria para cumplir con la misma. Esto permite que los sistemas tiendan a ser más fáciles de entender, mantener y ampliar; y además presenta la oportunidad de reutilizar los componentes en futuras aplicaciones.

Beneficios (37):

- Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece el hecho de tener sistemas más robustos y de fácil mantenimiento.
- El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases "sencillas y más cohesivas que son más fáciles de comprender y de mantener.

Alta Cohesión

Asigna una responsabilidad de modo que la cohesión siga siendo alta. La alta cohesión permite que la clase que tiene una responsabilidad pueda interactuar con otras clases para poder realizar las tareas encomendadas.

Beneficios (37):

- Mejoran la claridad y la facilidad con que se entiende el diseño.
- Se simplifican el mantenimiento y las mejoras en funcionalidad.
- A menudo se genera un bajo acoplamiento.
- La ventaja de una gran funcionalidad soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico.

Controlador

Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase (37). Es decir el controlador es el encargado de manejar un evento del sistema.

Singleton

Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. En Grails todos los servicios son Singleton ya que existe una instancia de la clase que se inyecta en todos los artefactos que declaren la variable correspondiente (29).

3.2.5 Diagrama de clases del diseño

El diagrama de clases del diseño no es más que una representación de clases y asociaciones. Los tipos de clases que se utilizan describirán la relación que tendrán al momento de la ejecución de la aplicación y cómo será su comunicación entre ellas.

A continuación se muestra el diagrama de clase de diseño del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia.

Diagrama de clases del diseño del subsistema Almacenamiento y Conservación.

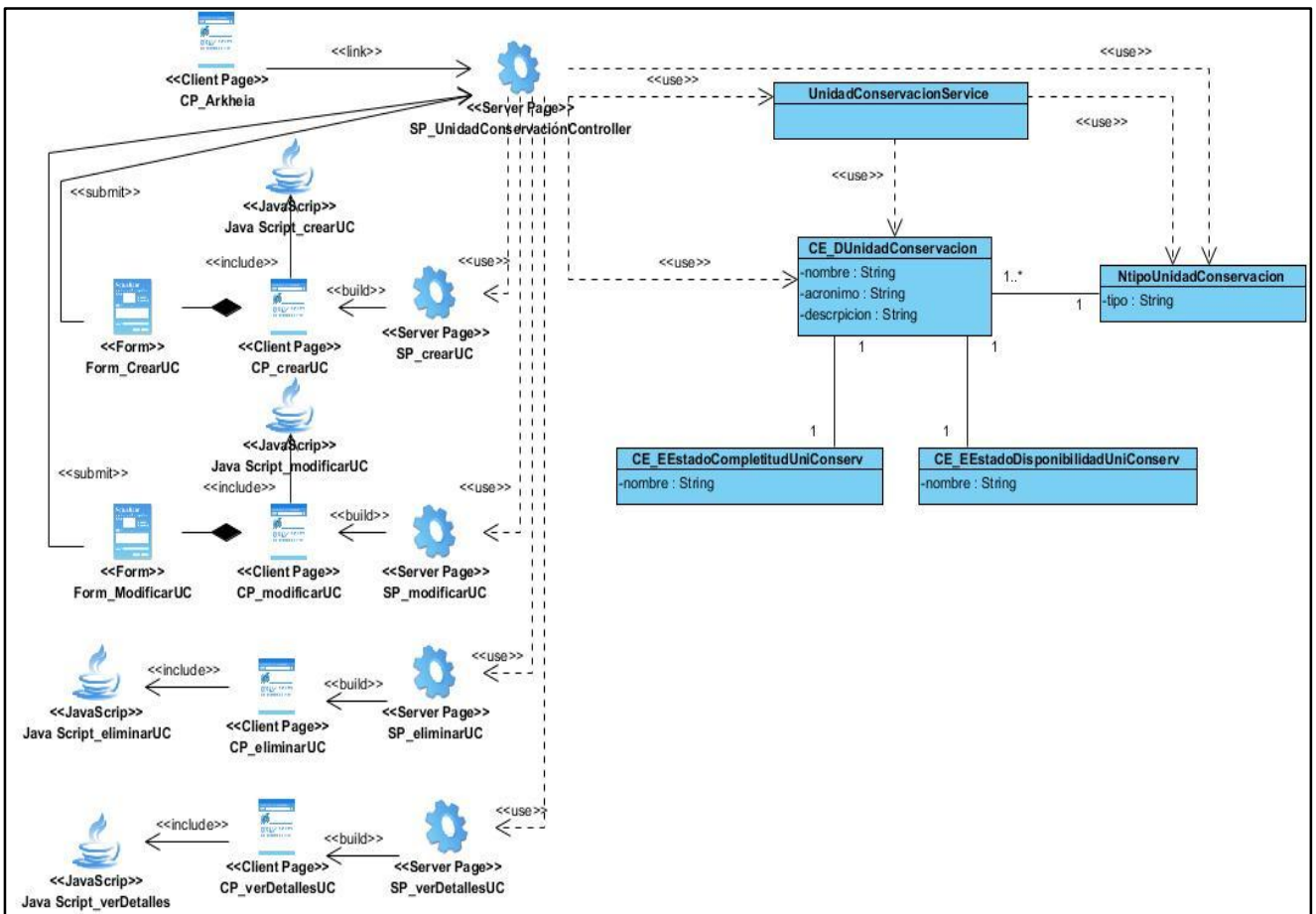


Figura.15 CU Gestionar Unidad de Conservación del módulo Almacenamiento.

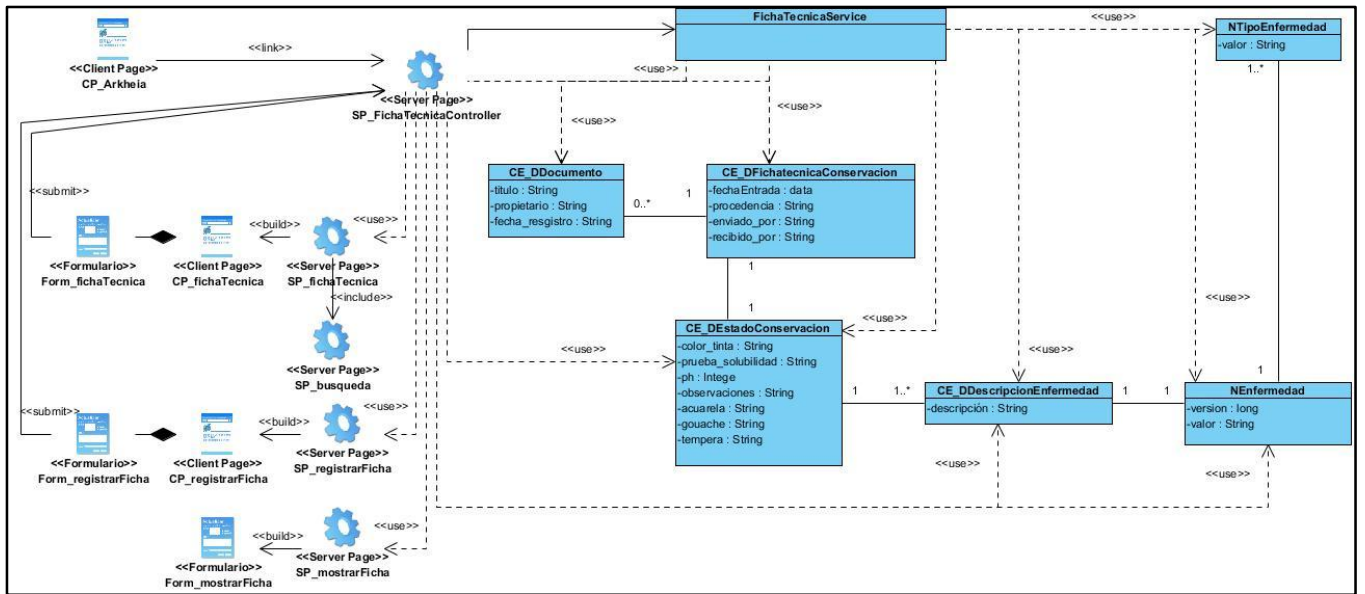


Figura.16 CU Gestionar Ficha Técnica de Conservación del módulo Conservación

3.2.6 Diagrama de clases persistentes

En el diagrama de clases persistentes se muestran las clases que permanecerán aun después de terminado la ejecución del software.

Diagrama de clases persistentes del módulo Almacenamiento

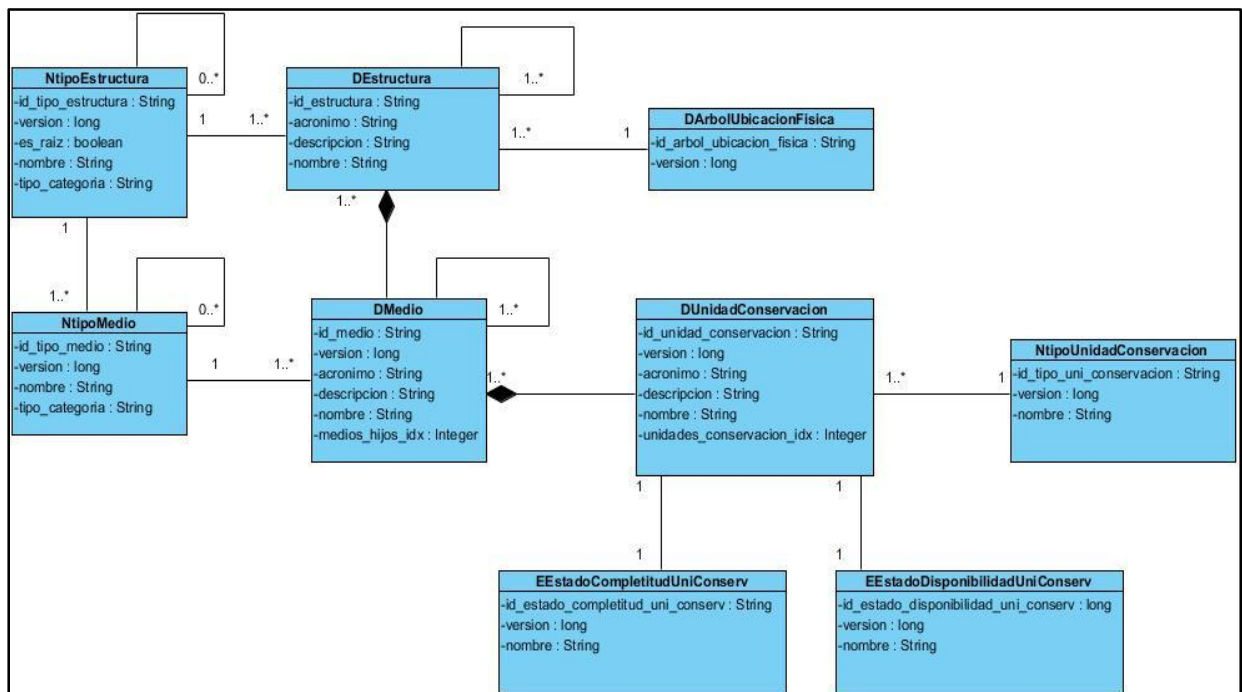


Figura.17 Diagrama de clases persistentes del módulo Almacenamiento.

DEstructura: La estructura (pertenece a un tipo de estructura) es el local que se dispone para almacenar y conservar los archivos. La misma puede tener más estructuras dentro de esta.

NtipoEstructura: El tipo de estructura agrupa un conjunto de estructuras.

DMedio: Es donde se ubicaran los archivos organizadamente. Los medios (pertenecen a un tipo de medio) pueden estar compuestos por más medios.

NtipoMedio: El tipo medio agrupa un conjunto de medios.

DUnidadConservacion: La unidad de conservación (pertenece a un tipo de unidad de conservación) es donde estarán almacenados los archivos.

NtipoUnidadConservacion: El tipo de unidad de conservación agrupa un conjunto de unidades de conservación.

EEstadoCompleitudUniConserv: Informa que la unidad está completa es decir ya no tiene su capacidad máxima de archivos guardados.

EEstadoDisponibilidadUniConserv: Permite saber si una unidad de conservación está disponible.

DArbolUbicacionFisica: Visualiza las estructuras, medios e unidades de conservación donde están almacenados los archivos.

Diagrama de clases persistentes del módulo Conservación

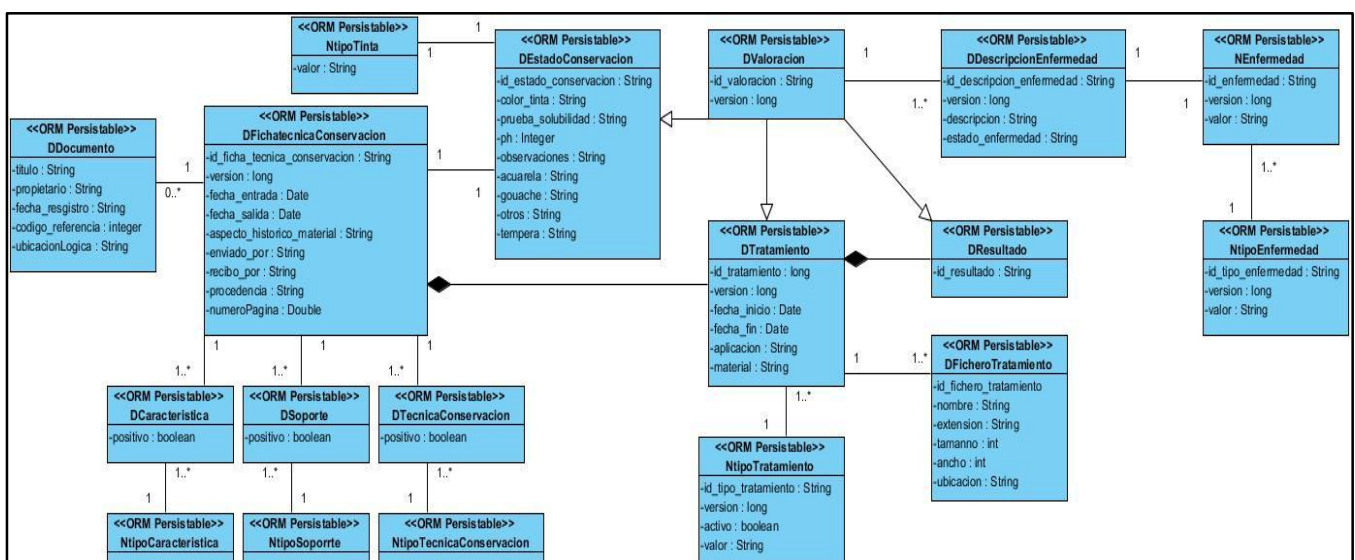




Figura.18 Diagrama de clases persistentes del módulo Conservación.

DDocumento: Contiene información que lo identifica, como la fecha de creación, título, productor entre otros aspectos.

DFichatecnicaConservacion: Contiene información sobre el diagnóstico técnico de los documentos de la oficina.

DValoración: El resultado, tratamiento y descripción de las enfermedades tienen una valoración.

EEstadoConservacion: Contiene información sobre el estado de conservación de los documentos.

DTratamiento: Contiene información sobre el tratamiento (pertenece a un tipo de tratamiento) que se le realiza a los documentos que así lo requieran.

NtipoTratamiento: El tipo de tratamiento agrupa un conjunto de tratamientos.

DResultado: Contiene información sobre el resultado después de aplicado el tratamiento.

DDescripcionEnfermedad: Describe la enfermedad que tiene el documento.

NEenfermedad: Contiene información sobre la enfermedad que tiene el documento (pertenece a un tipo de enfermedad).

NtipoEnfermedad: El tipo de enfermedad agrupa un conjunto de enfermedad.

DFichaRestEnc: Contiene información sobre la pieza (documento) que se le realiza la restauración y/o encuadernación.

NTipoPieza: Contiene información sobre el tipo de pieza.

DFicherosTratamiento: Contienen fotos antes del tratamiento y después del tratamiento realizado a la pieza.

3.5 Conclusiones

En el presente capítulo se realizó un resumen del diseño del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia. Se muestran un grupo de diagramas y una breve descripción de los patrones utilizados, los cuales facilitan un mejor entendimiento en el desarrollo de los procesos de negocio del subsistema. Se tomó como muestra un caso de uso de cada módulo del subsistema a los que se le realizó el diagrama de colaboración y de clase del diseño. Contiene el diagrama de paquete del subsistema y el diagrama de clases persistentes de los módulos Almacenamiento y Conservación.

Capítulo.4 Implementación y Prueba

4.1 Introducción

En el presente capítulo se propone explicar a partir de los resultados del diseño, la implementación del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia, integrado por los módulos Almacenamiento y Conservación, denotando mediante componentes y relaciones entre ellos, el estado actual en que se encuentra el subsistema. Como parte de la implementación se muestra, el diagrama de componentes, el cual contribuye a la organización del subsistema y representa la vista estática de este, y el diagrama de despliegue en el que se visualiza la situación física del sistema. Se definen las pruebas de software como elemento esencial que garantiza la calidad del subsistema y revisión final del cumplimiento de los requisitos, especificaciones de diseño y codificación.

4.2 Implementación

La implementación describe cómo los elementos del diseño se implementan en términos de componentes, cómo dependen unos de otros y cómo estos se organizan de acuerdo a los nodos específicos en el diagrama de despliegue.

4.2.1 Diagrama de componentes

Un diagrama de componente describe los elementos físicos del sistema y la relación o dependencias lógicas entre los componentes del software. Forma parte de la arquitectura de software aunque se puede utilizar para modelar y documentar cualquier arquitectura del sistema. El diagrama de componentes muestra parte de la vista física de un sistema, en el que se modela la estructura de implementación de la aplicación, su organización en componentes y su despliegue en nodos de ejecución.

A continuación se muestra una vista general de la composición del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia, por medio de los diagramas de componentes correspondientes, teniendo en cuenta la estructura de la implementación de la aplicación. En ellos se observa la relación de dependencia que tienen los módulos que componen el subsistema con otros componentes, teniendo en cuenta que los módulos necesitan de los demás componentes o viceversa para realizar determinada funcionalidad. Además, se observa en los diferentes diagramas, la existencia de componentes representados por el estereotipo “*subsistema*”, ejemplo el propio subsistema, los módulos, etc. Esto ocurre cuando se encuentran compuestos por otros componentes.

En el diagrama de la Figura.19, se muestra el subsistema como componente principal implementado, integrado por los componentes *Módulo Almacenamiento* y *Módulo Conservación* y la relación entre ellos, teniendo en cuenta que desde el módulo Conservación se puede explorar el árbol de ubicación física representado por el componente *ArbolUbicacionFisica* (Figura.20). La interfaz de los módulos es gestionada mediante el componente *Plantilla Xabal Arkheia*. Los módulos también se relacionan con el componente *TagLib*, librería de etiquetas que ayuda a componer funcionalidades para las interfaces, el componente Base Datos, usado para la gestión de datos en el sistema, y el componente *comun*, que contiene funcionalidades comunes para los módulos que integran el sistema Xabal Arkheia. Además, el módulo de Conservación usa el gestor documental Nuxeo, representado por el componente *Nuxeo*, mediante la interfaz de igual nombre para almacenar las fotos tomadas a los documentos cuando se realice el proceso de restauración y encuadernación.

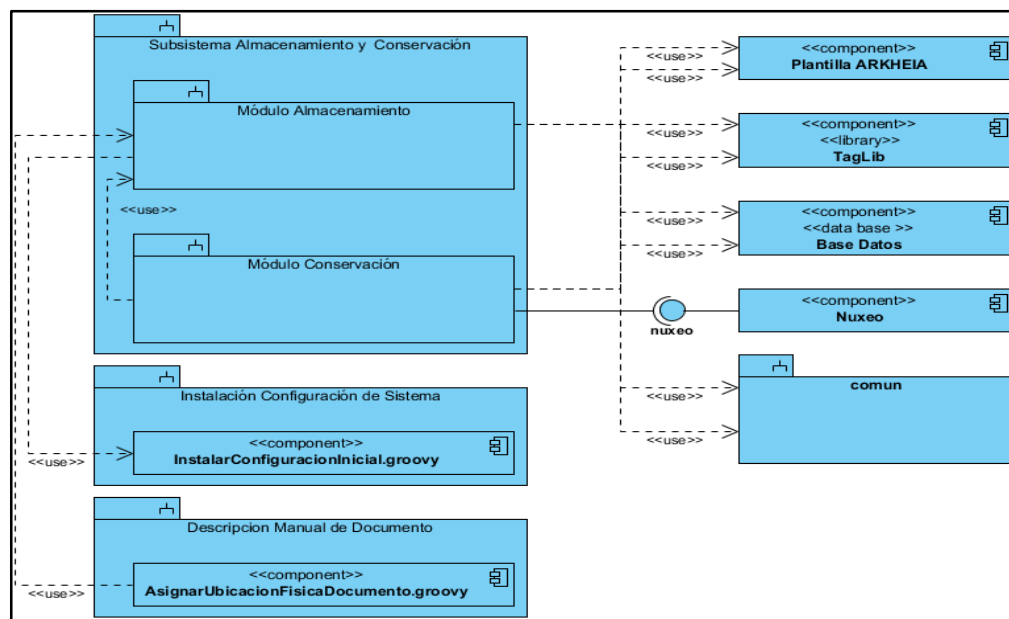


Figura.19 Diagrama de componente del subsistema Almacenamiento y Conservación del sistema Xabal Arkheia.

Otros componentes con los que se relaciona el subsistema, específicamente el módulo de almacenamiento, es con el componente *InstalarConfiguracionInicial.groovy* contenido dentro del componente *Instalación Configuración de Sistema*, ya que este define una estructura inicial para el funcionamiento del módulo. Además, el módulo de Almacenamiento permite explorar el árbol de ubicación

física para asignar una ubicación a un documento contenido en el componente *AsignarUbicacionFisicaDocumento.groovy*.

Los módulos del subsistema están constituidos por otros componentes (Figura.20 y Figura.21), que ayudan a desglosar y entender de mejor forma los procesos de almacenamiento y conservación. Los mismos, se encuentran conformados por el componente *Views* (vistas), que contiene las vistas o interfaces de usuario y son responsables de mostrar las distintas acciones disponibles, en el framework Grails son representadas por páginas servidoras de groovy (*.gsp*). Las vistas en el módulo de Almacenamiento están compuestas por los componentes *ArbolUbicacionFisica*, *EstructuraUbicacionFisica* y *UnidadesConservacion*. Mientras que el módulo de Conservación se compone por lo componentes *FichaTecnica*, ficha de restauración y encuadernación representado por *FichaResEnc*, *Aspecto*, *Tratamiento* y *Resultado*.

El componente *Controllers* (controladores), encargado de recibir los eventos desde las vistas, gestionar la aplicación de la lógica de negocio sobre el modelo de datos y determinar que vista debe mostrarse, en el framework Grails es representado por controladores (*controllers*). Los controladores en el módulo de Almacenamiento están compuestos por los componentes *ArbolUbicacionFisicaController.groovy*, *EstructuraUbicacionFisicaController.groovy* y *UnidadesConservacionController.groovy*, y la relación entre ellos, dado a que en el árbol de ubicación física se puede gestionar la estructura de ubicación física y la unidad de conservación, que a la vez pertenece a una estructura. Mientras que el módulo de Conservación se compone por lo componentes *FichaTecnicaController.groovy*, *FichaResEncController.groovy*, *AspectoController.groovy*, *TratamientoController.groovy* y *ResultadoController.groovy*, además de la relación entre ellos, ya que al realizar la ficha técnica de conservación o la ficha técnica de encuadernación y restauración a un documento, se puede controlar los daños del mismo mediante el aspecto, y posibilita aplicar algún tratamiento. Los controladores también utilizan los componentes *commands* y *utils* los cuales facilitan realizar el enlace de datos, conversión y la validación de estos.

El componente *Services* (servicios), es el encargado de implementar la lógica de negocio; por lo general cada controlador implementa su propio servicio. Los servicios en el módulo de Almacenamiento están compuestos por los componentes *ArbolUbicacionFisicaService.groovy*, *EstructuraUbicacionFisicaService.groovy* y *UnidadesConservacionService.groovy*. Mientras que el módulo

de Conservación se compone por los componentes *FichaTecnicaService.groovy*, *FichaResEncService.groovy*, *TratamientoControllers.groovy* y *ResultadoControllers.groovy*. El componente *Domain* (Dominio), encargado de gestionar los datos de la aplicación, es donde se encuentran las entidades o clases que persisten en la Base de Datos. El módulo Almacenamiento por ejemplo contiene las entidades *DEstructura*, *DMedio*, *DUnidadConservacion*. Por su parte el módulo Conservación entre las que contiene se encuentran *Dfichatecnica*, *DDanno*, *DTratamiento*.

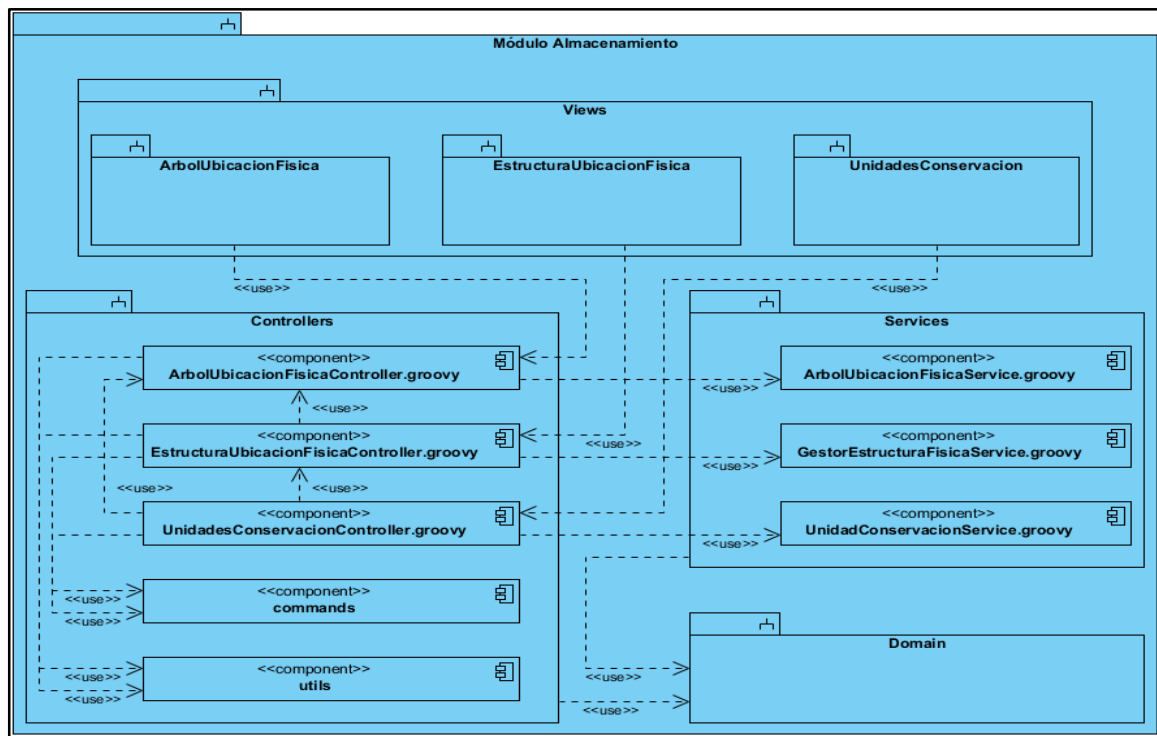


Figura.20 Diagrama de componente del módulo Almacenamiento.

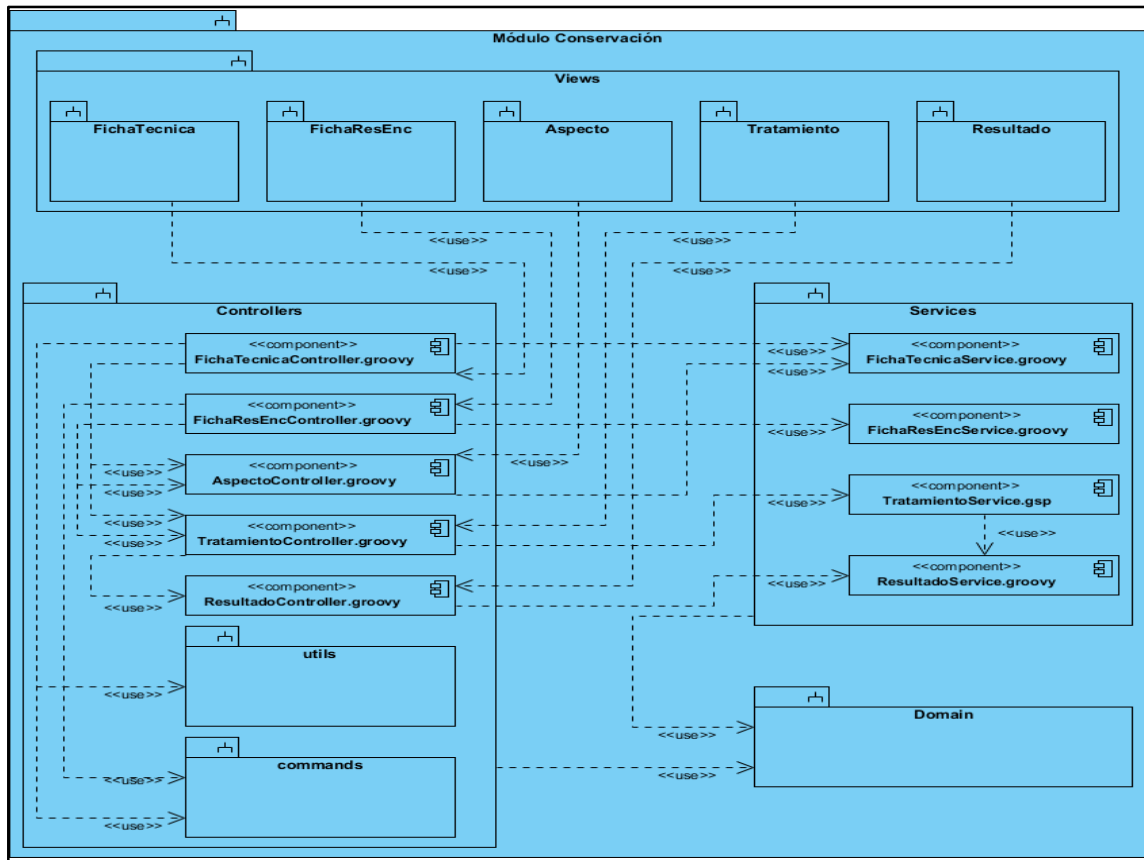


Figura.21 Diagrama de componente del módulo Conservación.

4.2.2 Diagrama de despliegue

Un diagrama de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos. Es el encargado de la distribución, entrega, instalación de las partes que constituyen el sistema físico.

Componentes que interactúan en el despliegue del sistema:(ver Figura.22)

- **PC Cliente_Usuario Externo:** Se emplea para las funciones de un usuario externo.
- **PC Cliente_Usuario Interno:** Es utilizada por los funcionarios de la institución y además se emplea para llevar a cabo la realización de los servicios que se le solicitan a la institución.
- **Servidor de Aplicaciones Web:** Atiende las peticiones de los usuarios y le proporciona la respuesta.
- **Servidor de Base de Datos:** Se almacenan todos los datos del sistema Xabal Arkheia.

- **Gestor de Documento Nuxeo:** Se almacenan los ficheros de los documentos.
- **Impresora_Multifuncional:** La impresora será multifuncional, ya que tendrá scanner e impresora para realizar los procesos de escaneo e impresión de los documentos.

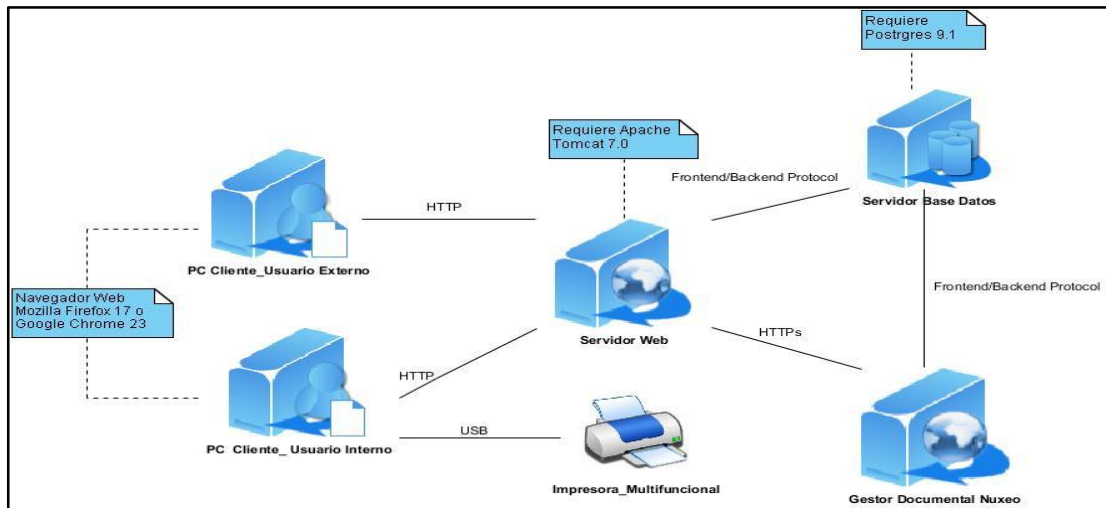


Figura.22 Diagrama de despliegue del sistema Xabal Arkheia.

4.3 Prueba

La prueba es un proceso de ejecución de un programa con la intención de descubrir errores. Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Estrategia de prueba utilizada:

Nivel de prueba. Prueba de Sistema: Son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados (30).

Tipos de Pruebas:

- **Funcionalidad:** Pruebas fijando su atención en la validación de las funciones, métodos, servicios, caso de uso (30).
- **Stress:** Enfocada a evaluar cómo el sistema responde bajo condiciones anormales (extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible) (30).

- **Usabilidad:** Prueba enfocada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto y en línea, asistente documentación de usuarios y materiales de entrenamiento (30).

Método de prueba. Prueba de Caja Negra: Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto (30).

El nivel de prueba de sistema incluye los tipos de prueba de funcionalidad, stress y usabilidad, con el uso de diseños de casos de prueba, utilizando el método de prueba de caja negra.

Caso de prueba está compuesto por un conjunto de entradas con datos de prueba, unas condiciones de ejecución, y unos resultados esperados. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces. El propósito de estos es identificar y comunicar las condiciones que se llevarán a cabo en la prueba.

4.3.1 Pruebas realizadas al subsistema Almacenamiento y Conservación

En esta sesión se realiza una descripción de cómo fueron realizadas las pruebas a los módulos del subsistema Almacenamiento y Conservación, además de mostrar los resultados de las mismas.

Cuando se terminaron las descripciones de los casos de usos de cada módulo se comenzaron a realizar el diseño de los casos de prueba para cada uno de estos casos de uso.

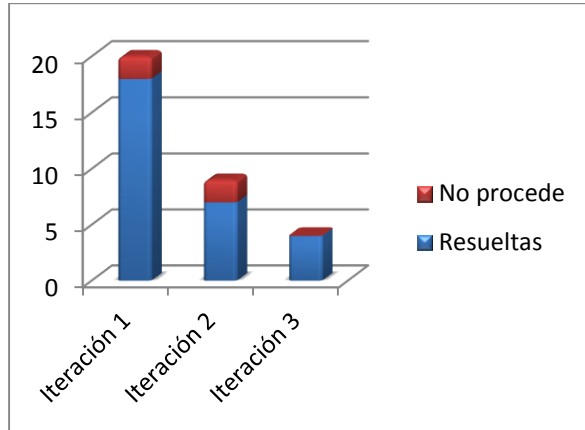
Implementados los módulos del subsistema y diseñados los casos de prueba, se comenzó a probar la aplicación, para verificar si se cumplieron los requisitos establecidos en los casos de usos. Para esto se realizaron las pruebas de liberación donde se ejecuta cada caso de prueba profundamente, se detectan las no conformidades⁸, y a partir de estas no conformidades encontradas se establecen varias iteraciones de pruebas.

Se realizaron pruebas de liberación por el equipo de calidad del centro ISEC y por el Centro Nacional de Calidad de Software (CALISOFT). Se hicieron tres iteraciones para el módulo de Almacenamiento y el módulo de Conservación del subsistema.

⁸**No conformidades:** Son problemas detectados en los casos de uso o del sistema.

Pruebas de liberación realizadas por el centro ISEC:

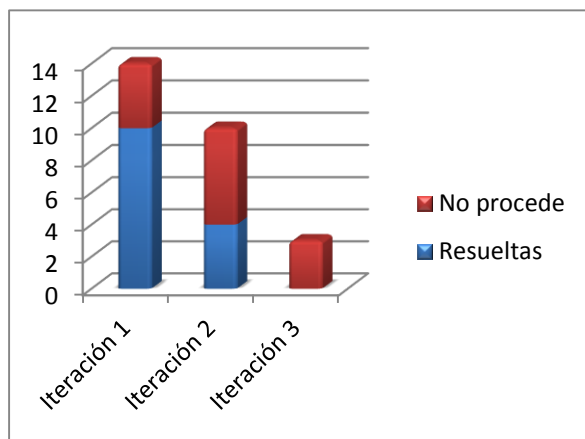
Resultados de las pruebas de liberación aplicadas al Subsistema Almacenamiento y Conservación:



No.Iteraciones	Resueltas	No procede	Total NC
1ra Iteración	18	2	20
2da Iteración	7	2	9
3da Iteración	4	0	4

Tabla.1 Resultados de las pruebas de liberación aplicadas al módulo Almacenamiento.

Gráfico.1 Resultados de las pruebas de liberación aplicadas al módulo Almacenamiento.



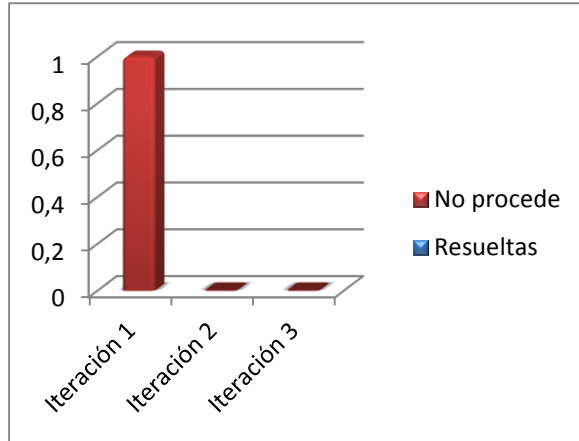
No.Iteraciones	Resueltas	No procede	Total NC
1ra Iteración	10	4	14
2da Iteración	4	2	6
3da Iteración	0	3	3

Tabla.2 Resultados de las pruebas de liberación aplicadas al módulo Conservación.

Gráfico.2 Resultados de las pruebas de liberación aplicadas al módulo Conservación.

Pruebas de liberación realizadas por CALISOFT:

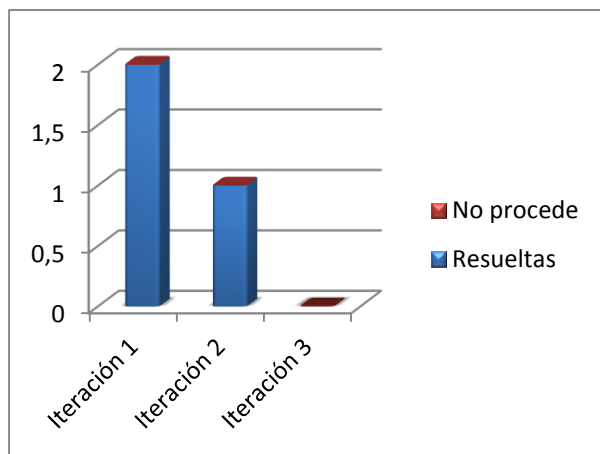
Resultados de las pruebas de liberación aplicadas al Subsistema Almacenamiento y Conservación:



No.Iteraciones	Resueltas	No procede	Total NC
1ra Iteración	0	1	1
2da Iteración	0	0	0
3da Iteración	0	0	0

Tabla.3 Resultados de las pruebas de liberación aplicadas al módulo Almacenamiento.

Gráfico.3 Resultados de las pruebas de liberación aplicadas al módulo Almacenamiento.



No.Iteraciones	Resueltas	No procede	Total NC
1ra Iteración	2	0	2
2da Iteración	1	0	1
3da Iteración	0	0	0

Tabla.4 Resultados de las pruebas de liberación aplicadas al módulo Conservación.

Gráfico.4 Resultados de las pruebas de liberación aplicadas al módulo Conservación.

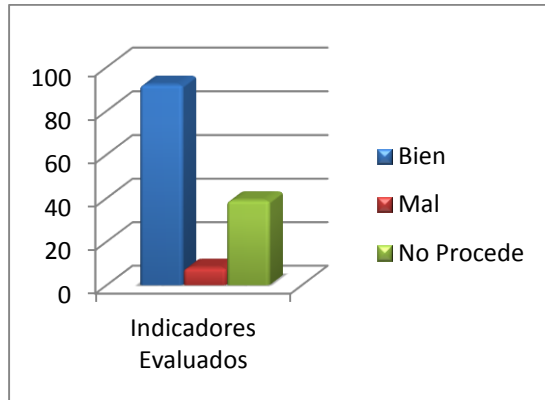
Pruebas de Stress y Usabilidad realizadas por CALISOFT:

Las pruebas de stress realizadas al subsistema arrojaron resultados positivos, demostrando que el sistema responde bajo condiciones anormales (extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible).

Las pruebas de usabilidad se realizaron a través de una lista de chequeo donde se evaluaron indicadores que se agruparon en varios aspectos como la visibilidad del sistema, adecuación del sistema, control y libertad de usuarios, consistencia y estándares, prevención de errores, ayudar a los usuarios a reconocer,

diagnosticar y recuperarse de errores, ayuda y documentación, accesibilidad, comprobaciones técnicas, semántica de la aplicación. Los indicadores se evaluaron de uno (1) en caso de que este mal, cero (0) en caso de que el elemento revisado no presente errores y en no procede el cual especifica que el indicador a evaluar no se puede aplicar en ese caso.

Resultados de las pruebas de Usabilidad



Total de indicadores	Total de Evaluaciones Bien	Total Evaluaciones Mal	No procede
140	92	8	39

Tabla.5 Resultados de las pruebas de Usabilidad.

Gráfica.5 Resultados de las pruebas de Usabilidad.

Al realizar las pruebas de liberación al subsistema en cuestión, se pudo comprobar que las no conformidades encontradas no afectaron el flujo básico de eventos y por lo tanto no alteraron el resultado final esperado.

4.3 Conclusiones

En el presente capítulo se muestra como se han desarrollado los módulos de almacenamiento y conservación del Xabal Arkheia, reflejado a través de los diagramas de componentes y cómo interactúa este con los demás componentes de la aplicación, además del diagrama de despliegue, el cual muestra cómo está constituida la distribución del sistema físicamente. Para lograr el buen funcionamiento del subsistema, se aplicaron pruebas de calidad de Caja Negra, de Stress y de Usabilidad a los módulos Almacenamiento y Conservación del subsistema.

Conclusiones

El presente trabajo culmina dando cumplimiento al objetivo trazado, de desarrollar el subsistema Almacenamiento y Conservación del sistema Xabal Arkheia, que facilitará la ubicación física de los documentos y disminuirá el deterioro de los mismos, para agilizar los procesos de almacenamiento y conservación de los documentos de los archivos históricos.

Se efectuó una fundamentación del estudio de los diferentes conceptos que forman parte de la gestión de documentos en los archivos históricos logrando un mejor entendimiento de los procesos de almacenamiento y conservación de los documentos de archivos.

El estudio de las tecnologías y herramientas definidas por el proyecto permitió determinar donde serían utilizadas efectivamente para el desarrollo de la solución propuesta.

Se definió el diseño que modela el subsistema y brinda soporte a los requisitos funcionales y no funcionales, siendo vital en la confección y concepción del diseño la utilización de patrones que permitieron aplicar buenas prácticas y brindar soluciones probadas a problemas comunes.

Se estructuró un diagrama de componentes que representa la implementación del sistema en términos de componentes, quedando construida una aplicación capaz de garantizar rapidez y efectividad en los procesos almacenamiento y conservación de los documentos de archivos.

Se realizó el diseño de las pruebas necesarias para comprobar el correcto funcionamiento de la aplicación. El diseño de pruebas obtenido fue utilizado en la validación del sistema, el cual comprobó la validez del mismo.

Recomendaciones

Con el propósito de enriquecer la propuesta de solución realizada, se recomienda:

- Flexibilizar el proceso de ubicación física, el cual permita crear la unidad de conservación y almacenar los documentos en ella, sin la necesidad de ubicarla en la estructura de ubicación física.
- Incluir los procesos de transferencia de documentos para que la aplicación pueda ser utilizada en entidades que trabajen con cualquier tipo de archivo.
- Llevar el control del registro de los parámetros ambientales garantizando que las estructuras y los medios tengan las condiciones adecuadas para la conservación de los documentos de los archivos.

Bibliografía

1. **Lododini, Elio.** *Archivística, Principios y Problemas.* Madrid : Anabad Castilla-La Mancha, 1993.
2. **Muñoz, Cesar Gutierrez.** *Archivística.* Perú, Lima : s.n., 1991.
3. **Heredia, Antonia Herrera.** *Archivística General Teoría y Práctica.* Sevilla, España : EXCMA.Diputación Provincial de Sevilla, 1991.
4. **Martinic, Beatriz Mercado.** *Guía para el Diseño e Implementación de un Sistema de Gestión de Archivos.*
5. **Morales, Gloria.** *Los Documentos Históricos.* Venezuela, Zulia : s.n., junio, 2010.
6. **Pacheco, Juan Ramón Romero Fernández.** *Conservación y Reproducción.* Madrid, España : Ministerio de Educación y Cultura, 1999.
7. **ADOC.** ADOC Administración de Documentos LTDA. [En línea] 2011. [Citado el: 20 de enero de 2013.] <http://www.adoc.com.co>.
8. Catálogo de Software.com. [En línea] [Citado el: 20 de enero de 2013.] <http://www.catalogodesoftware.com/producto-adoc-ltda-1114>.
9. **ADOC.** Blog ADOC LTDA. [En línea] [Citado el: 20 de enero de 2013.] Sitio: <http://adocltida.blogspot.com/2010/05/bienvenidos-al-blog-de-adoc-ltda.html>.
10. Infopedia. [En línea] [Citado el: 20 de enero de 2013.] <http://infopedia.com.co/portafolio-de-servicios/herramientas-de-gestion/docunet-gestion-documental/>.
11. Catálogo de Software.com. [En línea] [Citado el: 20 de enero de 2013.] Sitio: <http://www.catalogodesoftware.com/producto-docunet-8126>.
12. **SEPG.** Administración presupuestaria. [En línea] [Citado el: 20 de enero de 2013.] <http://www.oficinavirtual.pap.minhap.gob.es/sitios/oficinavirtual/ES-ES/CATALOGOSISTEMASINFORMACION/DOCUNET/Paginas/MenuSitio.aspx>.
13. Catálogo de Software.com. [En línea] [Citado el: 20 de enero de 2013.] Sitio: <http://www.catalogodesoftware.com/producto-sade-net-8127>.
14. Guía de soluciones TIC. [En línea] [Citado el: 20 de enero de 2013.] <http://www.guiadesolucionestic.com/sistemas-de-informacion/gestion-documental/administracion-y-gestion-electronica-de-documentos/617-siadocr->.
15. **SIADOC.** Siscomputo. [En línea] [Citado el: 20 de enero de 2013.] <http://www.siscomputo.com/aplicaciones/web/portalweb.nsf/0/FA0A32AB7E5F22800525761300583B5F?OpenDocument>.

16. **Díaz, Lic. Fernando Antonio Ávila.** La Gestión Documental en Cuba. Resultados de la Política Cultural de la Revolución. Cuba, Santiago de Cuba : MEGACEN, 2007.
17. **Cultura, Ministerio de.** Archivo Histórico Provincial Villa Clara. [En línea] 5 de mayo de 2009. [Citado el: 17 de enero de 2013.] <http://archivohistorico.villaclara.cu/informaciones/documentos/Decreto-ley-265>.
18. **Vilma Quispe, Harry Huamantuca, Jose.L Vargas.** Metodología RUP (Rational Unified Process). Puno : Universidad Nacional del Altiplano, 2011.
19. Visual Paradigm. UML, BPMN and Database Tool for Software Development. [En línea] [Citado el: 21 de enero de 2013.] <http://www.visual-paradigm.com>.
20. Sitio de descargas de software. [En línea] [Citado el: 21 de enero de 2013.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
21. Mastermagazine. [En línea] [Citado el: 21 de enero de 2013.] <http://www.mastermagazine.info/termino/5234.php#ixzz2MWgloVQG>.
22. SpringSource. [En línea] [Citado el: 21 de enero de 2013.] <http://www.springsource.com/developer/sts>.
23. TortoiseSVN. Un cliente de Subversion para Windows Versión 1.7. [En línea] [Citado el: 21 de enero de 2013.] <http://es.scribd.com/doc/77058028/TortoiseSVN-1-7-3-es>.
24. **Machín, Katia Almeida.** *Procedimiento para la prestación de servicios por la Comunidad Técnica Cubana de PostgreSQL.* Cuba, La Habana : UCI. Trabajo de Diploma, junio de 2011. TD_04575_11.
25. Centro de Documentación de Nuxeo. [En línea] Open Source. [Citado el: 21 de enero de 2013.] <http://doc.nuxeo.com/display/MAIN/Nuxeo%2BDocumentation%2BCenter%2BHome%3Bjsessionid%3D4CBEE5D8E00B6FE17179DFA806C9CEDA&usg=ALkJrhhHT41LHWWJKq9I1WsTKQy0QAKWLQ>.
26. Apache Tomcat. [En línea] The Apache Software Foundation, 1999-2013. [Citado el: 21 de enero de 2012.] <http://tomcat.apache.org/>.
27. **Prof. Manuel Sánchez Valent.** Introducción a la Tecnología. [En línea] [Citado el: 21 de enero de 2013.] <http://platea.pntic.mec.es/~msanch2/tecnoweb/introduc.htm#Concepto%20y%20definici%C3%B3n>.
28. **Rebecca Murphey.** Fundamentos de jQuery. [En línea] Creative Commons Attribution-Share Alike. [Citado el: 21 de enero de 2013.] <http://librojquery.com/>.
29. **Brito, Nacho.** *Manual de desarrollo web con Grails.* España : Creative Commons Reconocimiento-No comercial-Sin Obras derivadas España, 9 de junio del 2009. ISBN: 978-84-613-2651.
30. **Ivar Jacobson, Grady Booch ,James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software.* España : Addison Wesley, 2000.

31. **Txema Rodríguez.** Genveta:dev Desarrollo y Software. [En línea] 16 de junio de 2012. [Citado el: 21 de enero de 2013.] <http://www.genbetadev.com/frameworks/bootstrap>.
32. Profesor Java. [En línea] 15 de junio de 2010. [Citado el: 21 de enero de 2013.] <http://profejvaoramas.blogspot.com/2010/09/lenguaje-unificado-de-modelado-uml.html>.
33. **Yaidel Ferrales Obregón, Dariel López Rodríguez.** *Módulos Incidencias e Indisciplinas y medidas disciplinarias del SIDE P.* Cuba, La Habana : UCI. Trabajo de Diploma, 2012. TD_05338_12.
34. **Javier García de Jalón, José Ignacio Rodríguez y otros.** *Aprenda Java como si estuviera en primero.* San Sebastián : Empresa Informatica, enero, 2000. ISSN: 0000044234.
35. Joonla-Chile.cl Desarrollo de tecnología web avanzada. [En línea] TILATAM.com. [Citado el: 21 de enero de 2013.] <http://joomla-chile.cl/diplomado-u-de-chile-joomla-chile/unidad-i/135-joomla-y-sus-lenguajes-de-programacion-php-html-xhtml-sql-xml-javascript-ajax-css.html>.
36. **Granados, Alejandro García.** Desarrollo Ágil de Aplicaciones Web con Grails Framework. enero, 2012.
37. **Lerman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado objeto.* Mexico : Prentice Hall, 1999. ISBN/970-17-0261-1.
38. **Aguilar, Violena Hernández.** Proceso de pruebas de caja negra basado en la descripción de casos de uso.
39. **Mugica, Mayra Mena.** El ciclo de vida y la teoría de las edades de los documentos. *Gestión Documental y Organización de Archivos.* La Habana : Félix Varela, 2005, 1, págs. 35-36.
40. **Martha M Ferriol, Olga M Pedierro, Marisol Mesa León, Mercedes Maza Llovet.** *Manual de procedimiento para el tratamiento documental.* Archivo Nacional de la República de Cuba. La Habana : s.n., 2013. pág. 32.
41. **Rossini, Daniel.** *Los archivos y las nuevas tecnologías de la información.* [En línea] septiembre 2003.
42. Archivo Nacional de la República de Cuba. [En línea] 2009. [Citado el: 25 de noviembre de 2012.] <http://www.arnac.cu/index.php/archivo-nacional>.
43. **Sánchez, MSc Iliana Regla Alfonso.** *Clasificación de documentos.* Cuba, La Habana : Ciencias Médicas, 2008. pág. 16.
44. **Departamento de Información Pública de las Naciones Unidas.** Glosario de términos de mantenimiento de expedientes: *Sección de gestión de archivos y expedientes de las Naciones Unidas.* [En línea] 2011. [Citado el: 19 de enero de 2013.] <http://www.un.org/spanish/archives/unrecordsmgmt/glossaryofrecordkp.shtml#n>.

- 45. Gómez, Alejandro Delgado.** *Normalización de la descripción archivística: Introducción a Encoded Archival Description (EAD)*. Cartagena : s.n., 2004.
- 46.** *Norma Internacional sobre los Registros de Autoridad de Archivos Relativos a Instituciones, Personas y Familias: ASAAR (CPF)*. [Documento] Madrid : s.n., 2004.
- 47. Cruz, José Ramón.** Manual de Archivística. [aut. libro] Cruz Mundet. *Manual de Archivística*. 2da. Madrid : Fundación Germán Sánchez Ruipérez, 10, págs. 230-231.
- 48.** *Modelo de Requisitos para la Gestión de Documentos Electrónicos de Archivo*. [Documento] 2001. ISBN:84-604-6312-5.
- 49. Martinic, Beatriz Mercado.** *Guía para el Diseño e Implementación de un Sistema de un sistema de gestión de Archivo*. Chile : s.n., 2011.
- 50. Gestión Documental.** *Quipux*. [En línea] 05 de 2009. [Citado el: 12 de diciembre de 2012.] <http://www.quipux.org.ec/>.
- 51. D, Jorge Gutiérrez.** *Sistema de Gestión de Archivos.Fundamentos Teóricos*. [Documento] Colombia : s.n., 2009.
- 52. León, Eduardo.** Tutorial Visual Paradigm for UML. *Tutorial Visual Paradigm for UML*.
- 53.** Open Source. (2001-2012). *Nuxeo. Enterprise Content Management*. Recuperado el 21 de enero de 2013, de <http://www.nuxeo.com/es>.
- 54.** DGCmedia. [En línea] [Citado el: 21 de enero de 2013.] www.dgcmedia.es/tecnologias-web/jquery.
- 55. López, Esteban Saavedra.** *Grails: Framework para el desarrollo de aplicaciones Web*. Bolivia : s.n.
- 56. Escuela de Groovy y Grails.** Groovy. [En línea] [Citado el: 21 de enero de 2013.] <http://groovy.codehaus.org/>.
- 57.** Conozca más sobre la tecnología Java. *Java*. [En línea] [Citado el: 21 de enero de 2013.] <http://www.java.com/es/about/>.
- 58. Pressman, Roger.** *“Ingeniería del Software. Un enfoque práctico”*.2002. s.l. : McGraw-Hill/Interamericana, 2002.
- 59. Pressman, Roger.** *Software Engineering: a practitioner’s approach*. Estados Unidos : Mc Graw Hill, 2005.

Glosario de Términos

Estructura: Edificación o espacio físico donde se ubican los medios de una institución.

Medio: Mobiliarios con que cuenta la estructura. Los medios son un conjunto de muebles u objetos que sirven para facilitar los usos y actividades habituales en las oficinas o locales de una entidad.

Nodo: Son los medios y estructuras que están ubicados en la estructura de ubicación física, puede contener otros nodos.

Categoría: Está compuesta por las estructuras o medios que defina una institución.

Unidad de Conservación: Es la unidad en la cual se almacenan los archivos de una institución.

Estructura de ubicación física: Está representada por un árbol de ubicación física, el cual visualiza las estructuras, medios y unidades de conservación de una entidad. Permite simplificar la búsqueda de los documentos contenidos en una unidad de conservación.

Ficha Técnica: Es un documento que contiene la descripción de las características de un archivo de manera detallada.

Tratamiento: Es un conjunto de medios que se utilizan para curar las enfermedades que contiene un documento.

Enfermedad: Son daños que atentan contra la vida de un documento.

- Ejemplos de enfermedades: páginas desgarradas, rotas, con fango entre otras.

Resultado: es el efecto, consecuencia o conclusión del tratamiento realizado al documento.

Usuario: Persona de la institución que cuenta con permisos para acceder a la aplicación.

Rol: Papel que desarrolla un determinado *usuario* dentro de la aplicación. Cuenta con una serie de permisos.

Caso de Uso: Secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.