



Universidad de las Ciencias
Informáticas

Facultad 2

Centro de Informatización de la Gestión Documental
(CIGED)

Configuración de la norma ISAD (G) para el Sistema XABAL Arkheia 3.0

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

Autor:

Alberto Danyer López Mederos

Tutores:

MSc. Madelis Pérez Gil
Ing. Roberto Soriano Sifontes

La Habana, 2017
“Año 59 de la Revolución”



**“Si he logrado ver más lejos, ha sido porque
he subido a hombros de gigantes”**

Isaac Newton

Declaración de autoría.

Declaramos ser el legítimo autor del trabajo de diploma titulado: "Configuración de la norma ISAD (G) para el sistema XABAL Arkheia 3.0", y reconocemos a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los __días del mes de _____del año ____.

Alberto Danyer López Mederos

MSc. Madelis Pérez Gil

Ing. Roberto Soriano Sifontes

Dedicatoria

A mis padres.

Agradecimientos

A mis padres por su amor, dedicación y sacrificio sin límites.

A mi novia por estar siempre.

A mis hermanas y hermanos por su apoyo.

A mis compañeros de aula por las dudas compartidas.

A mis tutores por su tiempo y su ayuda.

Resumen

XABAL Arkheia es un Sistema de Gestión de Archivos Históricos desarrollado por el Centro de Informatización de la Gestión Documental (CIGED) de la Universidad de las Ciencias Informáticas (UCI). Este sistema se basa en la aplicación de la Norma Internacional General de Descripción Archivística (ISAD (G)) para la elaboración de las descripciones de los documentos de archivo. Dicha norma propone la descripción multinivel de los fondos documentales. En su estado actual el sistema sólo permite la descripción por igual de los niveles que integran el fondo documental y las unidades de descripción simple y compuesta y no comprende la navegabilidad entre las descripciones relacionadas. La presente investigación está orientada a la configuración de la norma para los distintos niveles del fondo así como sus unidades de descripción, ya sean simples o compuestas, con el objetivo de cumplir con las reglas propuestas para la correcta descripción multinivel de los fondos documentales, mejorando así la experiencia de los usuarios del sistema XABAL Arkheia.

Palabras clave: descripción multinivel, fondo documental, norma, unidad de descripción, XABAL Arkheia.

Índice de Contenido

Introducción	11
Capítulo 1. Fundamentos Teóricos	14
1.1 Introducción	14
1.2 Conceptos Asociados	14
1.3 Estudio del estado del arte	15
1.3.1 Internacional	15
1.3.2 Nacional	16
1.3.3 Conclusiones del Estudio	17
1.4 Metodología de desarrollo de <i>software</i>	18
1.4.1 Proceso Unificado Ágil para los centros productivos en la UCI (AUP-UCI)....	18
1.5 Herramientas y lenguajes informáticos	20
1.5.1 Lenguajes de programación	20
1.5.2 Framework de desarrollo	21
1.5.3 Herramienta de modelado	21
1.5.4 Entorno de desarrollo integrado	21
1.5.5 Sistema gestor de bases de datos:	22
1.5.6 Herramienta de control de versiones:	22
1.6 Conclusiones del capítulo	22
Capítulo 2. Análisis y diseño de la solución propuesta	23
2.1 Introducción del capítulo	23
2.2 Propuesta de solución	23
2.3 Modelo conceptual.....	24
2.4 Requerimientos	25
2.4.1 Requisitos funcionales del sistema.....	25
2.4.2 Requisitos no funcionales del sistema.....	26
2.4.3 Técnicas de validación de requisitos	28
2.5 Modelo de Casos de Uso del Sistema	29
2.5.1 Actores.....	29
2.5.2 Patrones de Casos de Uso.....	29
2.5.3 Diagrama de Casos de Uso del Sistema	29
2.5.4 Especificaciones Casos de Uso	30
2.6 Diseño de la solución.....	46
2.6.1 Arquitectura del sistema	46

2.6.2 Patrones de diseño	47
2.6.3 Diagrama de clases del diseño	49
2.6.4 Validación del diseño propuesto.....	51
2.7 Conclusiones del Capítulo	57
Capítulo 3. Implementación y prueba	59
3.1 Introducción al capítulo	59
3.2 Diagramas de componentes	59
3.3 Diagrama de despliegue	62
3.4 Pruebas de <i>software</i>	62
3.4.1 Estrategia de pruebas	62
3.4.2 Pruebas unitarias	63
3.4.3 Pruebas de integración	63
3.4.4 Casos de prueba	64
3.4.5 Resultados de las pruebas aplicadas	66
3.5 Conclusiones del capítulo	66
Conclusiones Generales	67
Recomendaciones	68
Bibliografía Referenciada	69
Bibliografía Consultada	71

Índice de Figuras

Figura 1.1. Interfaz gráfica de Archeevo.....	16
Figura 1.2. Interfaz gráfica de XABAL Arkheia 3.0.	17
Figura 1.3. Segundo escenario de la metodología AUP-UCI.....	20
Figura 2.1. Esquema de la propuesta de solución (Elaboración propia).....	24
Figura 2.2. Diagrama del Modelo de Dominio. Configuración de la norma ISAD (G).....	25
Figura 2.3. Prototipo de Interfaz. Configuración de la norma ISAD (G).	28
Figura 2.4. Diagrama de casos de uso del sistema.....	30
Figura 2.5. Principales componentes de la arquitectura Grails.....	46
Figura 2.6. DCD. CU Configurar norma ISAD (G).	50
Figura 2.7. DCD. CU Describir nivel.....	51
Figura 2.8. Resultados de la métrica TOC.	53
Figura 2.9. Resultados de la métrica TOC. Atributo responsabilidad.....	53
Figura 2.10. Resultados de la métrica TOC. Atributo complejidad de implementación. .	54
Figura 2.11. Resultados de la métrica TOC. Atributo reutilización.....	54
Figura 2.12. Resultados de la métrica RC.....	55
Figura 2.13. Resultados de la métrica RC. Atributo acoplamiento.....	56
Figura 2.14. Resultados de la métrica RC. Atributo complejidad de mantenimiento.	56
Figura 2.15. Resultados de la métrica RC. Atributo cantidad de pruebas.....	57
Figura 2.16. Resultados de la métrica RC. Atributo reutilización.	57
Figura 3.1. Diagrama de paquetes de componentes del sistema.	59
Figura 3.2. Diagrama de componentes del sistema. Módulo Administración.....	61
Figura 3.3. Diagrama de componentes del sistema. Módulo Clasificación.	61
Figura 3.4. Diagrama de componentes del sistema. Módulo Procesamiento de Documentos.....	61
Figura 3.5. Diagrama de despliegue de la propuesta de solución.	62
Figura 3.6. Resumen de las pruebas unitarias y de integración realizadas.	66
Figura 3.7. Resultados obtenidos mediante la aplicación de pruebas funcionales.	66

Índice de Tablas

Tabla 2.1. Entidades del modelo de dominio.....	25
Tabla 2.2. Especificaciones. Servidor de aplicación.....	27
Tabla 2.3. Actores del sistema.....	29
Tabla 2.4. ECU CU 1. Configurar Norma ISAD (G)	32
Tabla 2.5. ECU CU 2. Registrar la descripción de un nivel de un cuadro de clasificación	36
Tabla 2.6. ECU CU 3. Ver detalles de la descripción de un nivel.....	38
Tabla 2.7. ECU CU 4. Revisar descripción.....	40
Tabla 2.8. ECU CU 5. Rectificar descripción desaprobadada.....	43
Tabla 2.9. ECU CU 6. Modificar descripción aprobada.....	45
Tabla 2.10. Relación entre Requisitos funcionales y Casos de Uso.....	46
Tabla 2.11. Relación entre el valor del TOC y los atributos de calidad que evalúa.....	52
Tabla 2.12. Rangos de valores para la evaluación técnica del TOC.....	52
Tabla 2.13. Umbrales para el TOC.....	52
Tabla 2.14. Relación entre el valor del RC y los atributos de calidad que evalúa.....	55
Tabla 2.15. Rangos de valores para la evaluación técnica del RC.....	55
Tabla 3.1. Escenario de prueba Modificar configuración ISAD (G) para unidades de descripción.....	65
Tabla 3.2. Escenario de prueba Modificar configuración ISAD (G) para unidades de descripción.....	65

Introducción

En la evolución de la humanidad cada generación se ha nutrido de la experiencia y el conocimiento de las generaciones precedentes. El estudio y la adaptación han sido herramientas fundamentales que han labrado el camino de la civilización. Esto solo ha sido posible gracias a invaluable objetos de diversos soportes conservados a través de los siglos. Es por esto que desde los mismos orígenes de la escritura esta necesidad dio surgimiento a los archivos históricos.

Los archivos son entidades responsables de organizar, describir, administrar y conservar a perpetuidad los documentos después de haber concluido su vida semi-activa y haber adquirido valores útiles para la investigación histórica, científica o social.

Si bien los archivos, como los documentos tienen una existencia clara y de larga duración, la ciencia que los estudia, por el contrario, es de factura relativamente reciente (Cruz Mundet 1996).

Diversos estudiosos del tema hacen coincidir los inicios de la archivística, como la disciplina encargada del arreglo de los archivos, con la publicación en 1898 del *Manual para la clasificación y descripción de los archivos* de los archiveros holandeses S. Muller, J. A. Feith, R. Fruin. Como otras disciplinas empíricas se ha desarrollado a base de observación y experiencia, dando como resultado su delimitación metodológica y conceptual, con un lenguaje propio que la identifica y distingue de otras ciencias o disciplinas afines.

Con la aparición, a partir de la década de 1980, de las llamadas Tecnologías de la Información y las Comunicaciones (TIC), la aceleración de su desarrollo trajo consigo el surgimiento de los Sistemas de Gestión Archivística (SGA). Además de brindar beneficios, han forzado a los archivos a repensar tanto su forma cómo sus métodos y han hecho patente una creciente preocupación por la internacionalización de las normas archivísticas. *En la medida que los archivos emplean de manera creciente tecnología informática y de redes para crear y mantener información esencial y valiosa, necesitan una razonable garantía de que la información que crean sobrevivirá a los rápidos cambios en hardware y software (...) Las normas de codificación independientes del hardware y del software ofrecen la única garantía razonable de información duradera (Pitti 1999).*

La gradual automatización de tareas que hasta hace pocos años se ejecutaban de forma manual ha dado paso a una necesaria optimización en el manejo de los datos así como en las vías de acceso a los mismos. El uso de las tecnologías proporciona mayor rapidez de acceso, comodidad, control y seguridad. En el caso de los archivos, han contribuido a cambiar la mentalidad y formación de sus investigadores (González García 1997).

La Universidad de las Ciencias Informáticas (UCI) es una de las instituciones cubanas esenciales por su contribución al desarrollo tecnológico y a la informatización de la sociedad. Combina las actividades docentes con actividades productivas. La misma cuenta con centros productivos especializados, entre los cuales se encuentra el Centro de Informatización de la Gestión Documental (CIGED) que está dedicado al desarrollo

de sistemas y servicios informáticos integrales de alta calidad y competitividad en la informatización o mejora de los procesos de gestión documental. Dicho centro cuenta con una línea de desarrollo de *software* dedicada a los sistemas de gestión de archivos históricos la cual tuvo su origen con el sistema ARCHIVENHIS, desarrollado para el archivo nacional de la República Bolivariana de Venezuela. ARCHIVENHIS brindaba reducida normalización de las descripciones de los documentos, cuestión que fue tomada en cuenta y solucionada con el desarrollo de Arkheia 1.0 para la oficina del conservador de la ciudad de trinidad y el valle de los ingenios. Todavía en Arkheia 1.0 se observan procesos no genéricos. XABAL Arkheia 2.0 surge como evolución, para adaptar los procesos a la Oficina de Asuntos Históricos del Consejo de Estado (OAHCE) y al Archivo Central de la UJC Nacional. La última versión liberada es XABAL Arkheia 3.0, evolución del sistema que permite la creación de procesos genéricos y por primera vez se registra el sistema como versión producto. El sistema XABAL Arkheia contempla la informatización de los procesos: consulta, conservación de documentos de archivo e incorporación.

El proceso de incorporación es el encargado de la clasificación, el procesamiento y el almacenamiento de los de documentos luego de su llegada al archivo. En dicho proceso la descripción de los documentos se realiza utilizando como guía la Norma Internacional General de Descripción Archivística (ISAD (G) por sus siglas en inglés). Dicha norma propone el concepto de descripción multinivel, entendido como un sistema jerarquizado de descripciones que debían cumplir cuatro reglas básicas:

1. Describir de lo general a lo particular.
2. Ofrecer la información pertinente de acuerdo con su nivel.
3. Vincular las descripciones.
4. No repetir la información.

Para el correcto cumplimiento de estas reglas la propia norma brinda la posibilidad de que los criterios que ella define sean utilizados o no, en cada tipo de nivel, de acuerdo a las necesidades de cada institución.

Al realizar un análisis del proceso de descripción que se desarrolla en la actualidad utilizando el sistema XABAL Arkheia 3.0, se constata que el mismo se realiza utilizando el criterio multinivel pero aún existen limitaciones en cuanto al cumplimiento de sus reglas:

- Se realiza la utilización de todos los elementos definidos por ISAD (G) para cada nivel del cuadro de clasificación sin posibilidad de elegir cuales utilizar o no.
- No resulta posible acceder desde la descripción de un nivel a la descripción de sus ancestros o hijos.

Teniendo en cuenta lo anterior expuesto se plantea el siguiente **problema a resolver**:
¿Cómo eliminar las limitaciones en la implementación de la norma ISAD (G) para garantizar una correcta descripción multinivel en el Sistema XABAL Arkheia 3.0?

Como **objeto de estudio** de esta investigación se tiene: Sistemas Informáticos para la Gestión de Información en fondos documentales.

Siendo el **objetivo general** de la presente investigación: Incorporar la configuración de la norma ISAD (G) para garantizar una correcta descripción multinivel en el Sistema XABAL Arkheia 3.0.

Se delimita como **campo de acción**: Sistema Informático para la Gestión de Información en fondos documentales XABAL Arkheia.

Para dar cumplimiento al objetivo general se trazaron los siguientes **objetivos específicos**:

1. Analizar el marco teórico conceptual de la investigación y realizar un estudio del estado del arte sobre otros sistemas, metodologías de desarrollo, tecnologías y herramientas actuales para la implementación del sistema.
2. Diseñar la solución propuesta para el sistema.
3. Implementar las funcionalidades del sistema.
4. Validar la solución propuesta mediante la aplicación de pruebas funcionales.

Con la finalidad de satisfacer estos objetivos, la presente investigación fue guiada por **métodos científicos**. Dichos métodos se rigen por protocolos que facilitan la obtención de conocimiento científico y resultados confiables, *la acción fundada sobre la teoría y la evaluación de las ideas a la luz de los hechos* (Bunge 2000).

Métodos teóricos empleados

Analítico-sintético

Estudia los hechos partiendo de la descomposición del objeto de estudio en cada una de sus partes para estudiarlas de forma individual.

Posibilitó la extracción de los elementos fundamentales relacionados con la gestión archivística y la aplicación de la norma ISAD (G) así como su criterio multinivel abriendo la puerta a conclusiones respecto a procesos.

Modelación

La modelación científica es una herramienta que permite al investigador reproducir el fenómeno que se estudia.

Facilitó la creación de esquemas para una mayor comprensión de los flujos de información en los procesos de gestión archivística, fundamentalmente en las tareas relacionadas con la descripción.

Métodos Empíricos

Análisis documental

Propone pautas para expresar la información, adquirida de fuentes bibliográficas, de un modo preciso y exponiendo las ideas fundamentales.

Permitió realizar con éxito un sumario de la bibliografía consultada, así como el análisis de la misma para dar solución a la problemática planteada.

Capítulo 1. Fundamentos Teóricos

1.1 Introducción

En toda investigación científica se hace necesario partir de un cúmulo de información conocida la cual compone el marco teórico que sustenta las soluciones propuestas. En el presente capítulo se realizará un acercamiento a conceptos medulares de la archivística, la Norma Internacional General de descripción Archivística, así como los sistemas de gestión documental. Además se abordará el estudio realizado a soluciones existentes que comparten características con la expuesta y se describirá el entorno de trabajo necesario para llevar a cabo el proceso de desarrollo de la propuesta de solución.

1.2 Conceptos Asociados

Documento

Información creada o recibida, conservada como información y prueba, por una organización o un individuo en el desarrollo de sus actividades o en virtud de sus obligaciones legales (“Norma ISO UNE 15489” 2006).

Fondo documental

Conjunto de documentos, con independencia de su tipo documental o soporte, producidos orgánicamente y/o acumulados y utilizados por una persona física, familia o entidad en el transcurso de sus actividades y funciones como productor (Consejo Internacional de Archivos 2000).

Norma de descripción archivística

Instrumento de seguridad para garantizar una alta y homogénea calidad en la descripción, con los beneficios que ello puede reportar en el trabajo interno de gestión de fondos (Steinmark 1996).

Norma Internacional General de descripción Archivística (ISAD (G))

Esta norma constituye una guía general para la elaboración de descripciones archivísticas. Permite la estandarización de las especificaciones elementales tanto a nivel nacional como internacional, independientemente del soporte físico o tipo documental de los documentos de archivo (Consejo Internacional de Archivos 2000).

La ISAD (G) determina la representación de la información utilizando 26 campos que combinados componen la descripción de una unidad archivística, los cuales se agrupan en 7 áreas. En cualquier descripción pueden utilizarse todos los elementos, contemplados en estas reglas generales aunque sólo una parte de ellos debe necesariamente utilizarse en todos los casos (Consejo Internacional de Archivos 2000).

Propone el concepto de descripción multinivel. Si se describe el fondo como un todo, éste debe representarse en una descripción. Si es necesario describir las partes que integran el fondo, éstas pueden describirse por separado. La suma total de todas estas

descripciones, jerárquicamente unidas entre sí, representa el fondo y las partes descritas (Consejo Internacional de Archivos 2000).

Gestión documental

Conjunto de principios, métodos y procedimientos tendientes a la planificación, manejo y organización de los documentos generados y recibidos por las organizaciones, desde su origen hasta su destino final, con el objeto de facilitar su utilización y conservación (Gaceta Oficial 2009).

Sistema de Gestión Archivística (SGA)

Son un tipo de *software* que comúnmente proveen soporte integrado para el flujo de trabajo archivístico (Spiro 2009).

Entre sus funcionalidades se encuentran, administración de colecciones, valoración, descripción, arreglo, conservación, publicación. Estos sistemas han aportado eficiencia a la comunidad archivística y han ayudado a tratar el problema de las colecciones ocultas.

1.3 Estudio del estado del arte

Un estudio del estado del arte permite *seguirle las huellas a un proceso hasta identificar su estado de desarrollo más avanzado. Es una forma de investigación que apoya otras estrategias también de investigación. Como resultado se tiene un conocimiento sobre la forma como diferentes actores han tratado el tema de la búsqueda, hasta dónde han llegado, qué tendencias se han desarrollado, cuáles son sus productos y qué problemas se están resolviendo* (Londoño Palacio, Maldonado Granados y Calderón Villafáñez 2014).

En la presente investigación dicho estudio se realizó para comprender el grado de desarrollo que han alcanzado los sistemas de gestión de archivos históricos. Para una mejor organización del mismo fue dividido en dos secciones de acuerdo a la procedencia de los sistemas, tanto del plano internacional como nacional.

1.3.1 Internacional

Archeevo

Es un *software* de gestión archivística capaz de manejar millones de registros y *terabytes* de contenido digital. Este *software* consta de nueve módulos funcionales que suplen las necesidades del personal archivístico más experimentado, descripción de registros, gestión de recursos digitales, publicación en línea, conservación y restauración, archivo intermediario, administración de depósitos, sala de referencia virtual, gestión de administración y productividad e interfaces interoperables programables (“Archeevo” 2016).

Se basa en un conjunto de normas internacionales cómo son la Norma Internacional General de Descripción Archivística (ISAD (G)), Descripción Archivística Codificada (EAD), Norma Internacional sobre los Registros de Autoridad de Archivos relativos a Instituciones, Personas y Familias (ISAAR (CPF)), Protocolo para la Minería de Metadatos (OAI-PMH), entre otras.

Es bien conocido por su atractivo diseño, alta adaptabilidad y escalabilidad, así como su mejora continua.



Figura 1.1. Interfaz gráfica de Archeevo.

Adlib Archive

Adlib Archive es un producto de la compañía Axiell ALM, que se encarga del desarrollo de *software* para el sector cultural. Ha sido implementado en cientos de instituciones alrededor del mundo. Su versatilidad lo hace ideal para el uso tanto en archivos organizacionales como gubernamentales.

Permite elegir entre las plataformas de bases de dato MS SQL Express, MS SQL Server y Oracle.

Entre las funcionalidades que soporta se encuentran:

- Procesamiento de colecciones.
- Indexado de nombres, sujetos y lugares.
- Administración de diccionarios.
- Uso investigativo.
- Referencias al material publicado.

Se encuentra disponible en tres versiones que difieren en términos de funcionalidad y flexibilidad:

1. Adlib Archive Standard incluye funcionalidades para la administración de bibliotecas.
2. Adlib Archive Plus permite integrar la administración de museos o bibliotecas.
3. Adlib Archive XPlus combina las funcionalidades de Adlib Archive, Adlib Library y Adlib Museum.

1.3.2 Nacional

Archivenhis

Sistema de gestión de archivos históricos desarrollado en la Universidad de Ciencias Informáticas (UCI) para el Archivo General de la Nación de Venezuela. Su principal

objetivo es la conservación y preservación de los archivos históricos que se almacenan en los Archivos, además de poder gestionar toda esta documentación a través de una aplicación Web. La funcionalidad esencial que brinda el sistema y que le suministra la información base es permitir a los archiveros realizar la descripción de los documentos según las norma ISAD (G) y mantener actualizados los niveles de organización de la documentación tanto físicos como lógicos. También se lleva el control sobre los préstamos de la documentación y los servicios que se brindan en la institución.

XABAL Arkheia 3.0

Aplicación multiplataforma que contribuye con la conservación y preservación de la memoria histórica de las instituciones de archivo. Arkheia es una aplicación informática que permite la gestión de los documentos en archivos históricos, al tiempo que facilita la incorporación, descripción mediante la norma ISAD (G) y conservación de los mismos. El sistema permite, también, la creación del cuadro de clasificación de la institución y registrar el almacenamiento físico de los documentos. También posibilita la informatización de los servicios de las instituciones de archivo (Manual de Usuario XABAL Arkheia 3.0 2014).

Gestiona los procesos de conservación y restauración de los documentos históricos archivando los momentos resultantes del tratamiento ejecutado sobre los mismos, así como los registros ambientales de los locales donde se conserven.

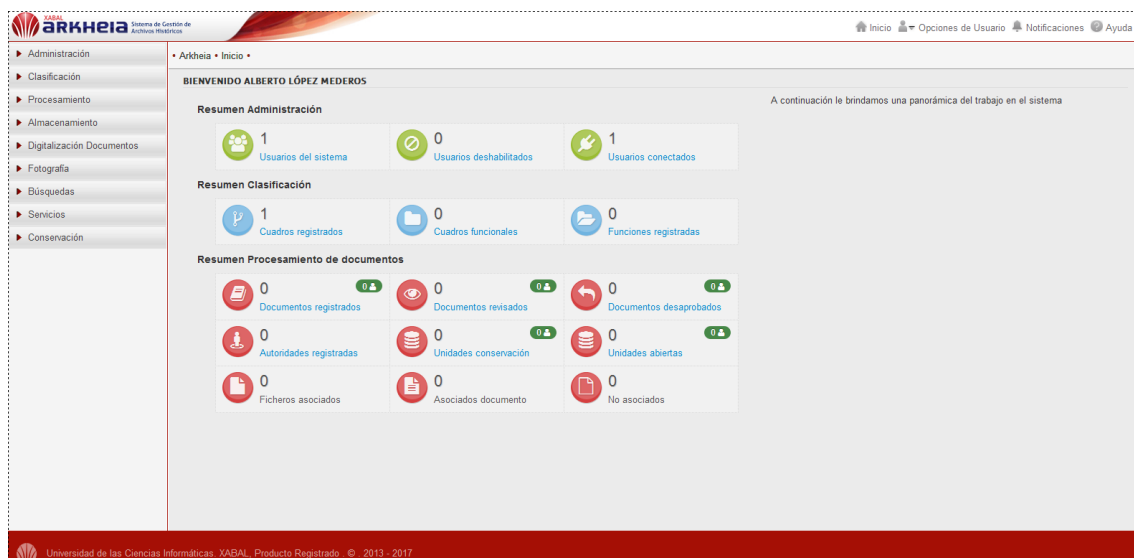


Figura 1.2. Interfaz gráfica de XABAL Arkheia 3.0.

1.3.3 Conclusiones del Estudio

Una vez realizado el estudio del estado del arte tanto en el plano internacional como nacional se arribó a los siguientes resultados.

- Adlib Archive ofrece soporte para estándares internacionales como ISAD (G). Adlib utiliza un cliente de escritorio basado en Windows y un servidor de base de datos. La publicación web de información archivística está disponible solo a través de la compra de Adlib Internet Server, el cual está construido sobre tecnologías Microsoft.

- El sistema Archeevo está basado en un conjunto de normas internacionales las cuales incluyen la ISAD (G) pero es un *software* propietario desarrollado por la empresa KeepSolutions.
- El Módulo para el Intercambio de Descripciones entre Sistemas de Gestión de Archivos Históricos Archivenhis permite describir documentos empleando la ISAD (G) y la creación de configuraciones para la misma. Sin embargo no permite la descripción multinivel del fondo documental.
- El sistema XABAL Arkheia en su versión 3.0 contempla la informatización de los procesos incorporación, consulta y conservación de documentos de archivo. Al realizar un análisis del proceso de incorporación resultan evidentes las limitaciones en la implementación del criterio multinivel. No resulta posible acceder a la información de los nodos padre desde sus hijos. De igual forma se realiza una utilización estática de todos los criterios definidos por ISAD (G) para cada uno de los niveles del cuadro de clasificación.

Se hace necesario emprender la mejora del proceso de incorporación en el sistema XABAL Arkheia 3.0 ya que en ningún caso las herramientas estudiadas satisfacen a cabalidad las necesidades existentes.

1.4 Metodología de desarrollo de *software*

Existen diversas propuestas metodológicas para el desarrollo de *software*. Estas se encuentran clasificadas en dos grandes grupos: tradicionales y ágiles. Las metodologías llamadas tradicionales, centradas fundamentalmente en el control del proceso de desarrollo mediante la definición rigurosa de las actividades, los artefactos que se deben producir y las herramientas a emplear. Las metodologías ágiles por su parte plantean centrarse en otros ámbitos, como pueden ser el factor humano o el producto de *software* en sí. Otorgando mayor valor al individuo, la colaboración con el cliente y al desarrollo incremental con ciclos de corta duración.

1.4.1 Proceso Unificado Ágil para los centros productivos en la UCI (AUP-UCI)

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP por sus siglas en inglés) es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- Desarrollo dirigido por Pruebas (test driven development - TDD en inglés).
- Modelado ágil.
- Gestión de cambios ágil.
- Refactorización de Base de Datos para mejorar la productividad.

Al no existir una metodología de *software* universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI (Rodríguez Sánchez 2015).

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que se llamará Ejecución y se agrega la fase de Cierre. Estas fases se subdividen en siete disciplinas, las cuáles se muestran a continuación.

- **Modelado de negocio:** Disciplina destinada a comprender los procesos de negocio de la organización.
- **Requisitos:** Disciplina que comprende la administración y gestión de los requisitos funcionales y no funcionales del producto.
- **Análisis y diseño:** En esta disciplina se modela el sistema y su forma para que pueda soportar todos los requisitos, incluyendo los requisitos no funcionales.
- **Implementación:** En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema.
- **Pruebas interna:** Se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas.
- **Pruebas de liberación:** Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa.
- **Pruebas de aceptación:** La prueba final antes del despliegue del sistema. Su objetivo es verificar que el *software* está listo y que puede ser usado por usuarios.

Escenario para la disciplina Requisitos

En la disciplina Requisitos de la metodología AUP-UCI los requerimientos pueden representarse mediante tres formas: casos de uso del sistema (CUS), historias de usuario (HU) y descripción de requisitos por proceso (DRP). Estas formas se agrupan en cuatro escenarios condicionados por el modelado de negocio.

Escenario No2:

La solución propuesta será desarrollada basándose en el segundo escenario que propone la metodología AUP-UCI que plantea:

Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el

objetivo primario es la gestión y presentación de información (Rodríguez Sánchez 2015).

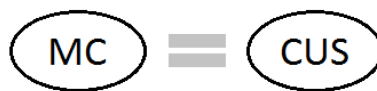


Figura 1.3. Segundo escenario de la metodología AUP-UCI.

1.5 Herramientas y lenguajes informáticos

Para la implementación de la propuesta de solución se hace necesario contar con tecnologías y herramientas informáticas establecidas por el líder del proyecto Archivo como política para el desarrollo del sistema XABAL Arkheia. Las cuales aseguran un entorno de trabajo apto para el desarrollo de dicha propuesta.

1.5.1 Lenguajes de programación

Groovy 2.0

Groovy es un lenguaje de programación multipropósito para la máquina virtual de Java (JVM). *Apunta a traer la expresividad de lenguajes como Ruby, Lisp y Python a la plataforma Java manteniéndose amigable con Java* (Keith Rocher 2006).

En su sintaxis presenta tantas similitudes con Java como diferencias, pero por encima de estas su meta principal es añadir prestaciones que faciliten las tareas comunes a las que se enfrentan los desarrolladores en Java.

JavaScript 1.8.5

Javascript es lo que se conoce como lenguaje script, es decir: se trata de código de programación que se inserta dentro de un documento. Javascript fue desarrollado por la empresa Netscape con la idea de potenciar la creación de páginas Web dinámicas para su navegador Navigator (Sánchez 2003).

Es un dialecto del estándar ECMAScript, basado en prototipos, imperativo, débilmente tipado y dinámico. Se emplea con mayor frecuencia en su forma del lado del cliente permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Su sintaxis es similar a la del lenguaje de programación C y adopta convenciones del lenguaje de programación Java.

Es probablemente el lenguaje de programación más abierto que existe: ECMA-262, su especificación, es un estándar ISO. Esa especificación es seguida de cerca por muchas implementaciones de partes independientes. Sus motores hecho enormes progresos, evolucionando desde lentos interpretes a compiladores rápidos y en tiempo. Ahora son lo suficientemente rápidos para la mayoría de las aplicaciones (Rauschmayer 2014).

1.5.2 Framework de desarrollo

Grails 2.2.1

Grails es un framework para desarrollo de aplicaciones web dinámicas sobre la plataforma de Java Enterprise Edition (JEE). Entre sus pilares fundamentales se encuentran: Groovy, que permite la creación de propiedades y métodos dinámicos en los objetos de la aplicación; Spring para la inyección de dependencias; Hibernate para la persistencia de los datos; SiteMesh para la composición dinámica de las vistas y Ant para la gestión del proceso de desarrollo.

1.5.3 Herramienta de modelado

Visual Paradigm for UML CE 8.0

Es una herramienta CASE con múltiples opciones para modelar diagramas UML2 y además soporta diagramas SysML y ER. La herramienta tiene un entorno de trabajo amigable, el cual facilita la manipulación y consulta de los proyectos de modelado. Es una herramienta de negocio y también posibilita realizar cambios específicos al código fuente de algunos lenguajes de programación como es el caso de C++ y Java.

Visual Paradigm también ofrece:

- Navegación entre la escritura del código y su visualización.
- Generador de informes en formato PDF/HTML.
- Ambiente visualmente superior de modelado.
- Sincronización de código fuente en tiempo real.

1.5.4 Entorno de desarrollo integrado

IntelliJ IDEA CE 14.0.1

Entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos desarrollado por la compañía JetBrains (anteriormente conocido como IntelliJ). Disponible en dos ediciones: edición comercial y edición comunitaria, esta última fue la elegida para la implementación de la propuesta de solución por ser de código abierto. Al momento de su primera liberación en el año 2001 fue uno de los primeros IDE de Java con avanzada navegación de código y capacidades de refactorización de código.

En el 2010 la revista Infoworld, IntelliJ recibió la puntuación de centro de prueba más alta fuera de las cuatro herramientas de programación superiores de Java: Eclipse, IntelliJ IDEA, NetBeans y Oracle JDeveloper.

Posee además soporte para lenguajes como Groovy y tecnologías como Grails, Ajax y Spring.

1.5.5 Sistema gestor de bases de datos:

PostgreSQL 9.1

Es un sistema de modelado objeto-relacional de código abierto basado en POSTGRES. Entre sus funcionalidades se encuentran, consultas complejas, llaves foráneas, disparadores, vistas e integridad transaccional.

Puede ser extendido por el usuario de diferentes maneras, añadiendo nuevos tipos de datos, funciones, operadores, métodos de indexado y lenguajes procedurales.

PostgreSQL utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

1.5.6 Herramienta de control de versiones:

Subversion 1.7

Subversion o SVN es un sistema de control de versiones (VSC por sus siglas en inglés) libre y de código abierto. Administra tanto archivos como directorios y los cambios efectuados a ellos a través del tiempo. Esto permite recuperar versiones antiguas de los datos o examinar el historial de cómo estos han cambiado. Soporta el trabajo en redes, lo que le permite ser usado por personas en computadoras diferentes al mismo tiempo.

Las principales características de SVN son:

- Mantiene versiones no sólo de archivos, sino también de directorios
- Mantiene versiones de los metadatos asociados a los directorios.
- Mantiene la historia de todas las operaciones de cada elemento, incluyendo la copia, cambio de directorio o de nombre.
- Atomicidad de las actualizaciones. Una lista de cambios constituye una única transacción o actualización del repositorio. Esta característica minimiza el riesgo de que aparezcan inconsistencias entre distintas partes del repositorio.

1.6 Conclusiones del capítulo

La recopilación de los conceptos fundamentales asociados al objeto de estudio permitió la elaboración del marco teórico. El cuál sirve como base para el desarrollo de la investigación.

El estudio del estado del arte mostró las características de los principales sistemas de gestión archivística a nivel nacional e internacional. Se hace evidente la necesidad de mejorar el proceso de incorporación en el sistema XABAL Arkheia 3.0 ya que en ningún caso se satisfacen a cabalidad las necesidades existentes.

Para la implementación de la propuesta de solución se hace necesario contar con herramientas y tecnologías de desarrollo establecidas como política del proyecto Archivo para el desarrollo del sistema XABAL Arkheia. Las cuales aseguran un entorno de trabajo apto para el desarrollo de dicha propuesta.

Capítulo 2. Análisis y diseño de la solución propuesta

2.1 Introducción del capítulo

En el presente capítulo se caracteriza la propuesta de solución brindada y se analizan las especificidades relacionadas con las disciplinas Modelación del Negocio y Requisitos. Se incluyen los artefactos derivados de dichas disciplinas, haciendo uso de los patrones de diseño adecuados. Además como parte de los procesos de diseño se definirá la arquitectura del sistema y se elaborarán los diagramas de clases del diseño, el cual será validado con la aplicación de métricas de calidad.

2.2 Propuesta de solución

Después de realizar un estudio de la problemática anteriormente planteada y teniendo como presente el uso del ambiente de trabajo definido para el desarrollo del sistema XABAL Arkheia 3.0, se propone como solución implementar la configuración de la norma ISAD(G).

Dicha configuración permitiría seleccionar los campos a utilizar durante la descripción de cada tipo de nivel en cualquier cuadro de clasificación, así como también para las unidades de descripción simple y compuesta. La solución incluiría ofrecer una navegación directa entre las descripciones relacionadas de acuerdo a su ubicación lógica, cumpliendo así con las pautas propuestas por la norma ISAD (G) para la realización de la descripción multinivel del fondo documental y facilitando el proceso de incorporación en el sistema XABAL Arkheia 3.0.

El sistema cuenta con cuatro subsistemas funcionales: Instalación y Configuración, Incorporación y Organización documental, Almacenamiento y conservación y Búsqueda y recuperación. En el subsistema funcional “Instalación y Configuración” se añadirá el módulo “Configuración de la norma ISAD (G)”. Dicho módulo permitirá selección de los elementos de descripción a emplear en los distintos niveles de los cuadros de clasificación. Elementos que posteriormente definirán las descripciones realizadas en el módulo “Procesamiento de documentos” del subsistema funcional “Incorporación y organización documental”. Como se ilustra en la figura 2.1.



Figura 2.1. Esquema de la propuesta de solución (Elaboración propia).

2.3 Modelo conceptual

La descripción del modelo conceptual tiene como objetivo representar el entendimiento común alcanzado por los involucrados respecto a los objetos y conceptos del dominio. Explica cuales son y cómo se relacionan los conceptos relevantes en la descripción del problema. Existen diversas variantes, con distintos grados de sofisticación, para describir un modelo conceptual. Considerando que la propuesta de solución reúne las características del segundo escenario de la metodología AUP-UCI y la evolución del sistema XABAL Arkheia se decide representar el modelo conceptual mediante un modelo de dominio.

El Modelo de dominio, descompone el espacio del problema en unidades comprensibles (conceptos). Puede verse como un modelo que comunica (a los interesados, como pueden ser los desarrolladores) cuáles son los términos importantes y cómo se relacionan (Larman 2003).

La figura 3 ilustra el Diagrama de Dominio de la propuesta de solución donde se relacionan las clases que se describen a continuación.

Clase	Descripción
Configuración ISAD(G)	Conjunto de campos de la norma ISAD (G) a utilizar en las descripciones.
Configuración ISAD(G) para nivel	Contiene los campos de la norma ISAD (G) usados para describir un tipo de nivel específico.
Configuración ISAD(G) para unidad de descripción	Contiene los campos de la norma ISAD (G) usados para describir las unidades de descripción simples o compuestas.
Descripción	Especificación de las características de un nivel del fondo documental empleando los elementos descriptivos que brinda la norma ISAD (G).
Nivel	Nodo del árbol de un cuadro de clasificación.
Notificación	Comunicación del resultado de una revisión a la

	descripción de un documento.
Revisión	Comprobación que se realiza a la descripción de un documento.
Unidad de Conservación	Medio donde se almacenan los documentos para su conservación física. Los documentos solo podrán ser almacenados en una única unidad de conservación.

Tabla 2.1. Entidades del modelo de dominio.

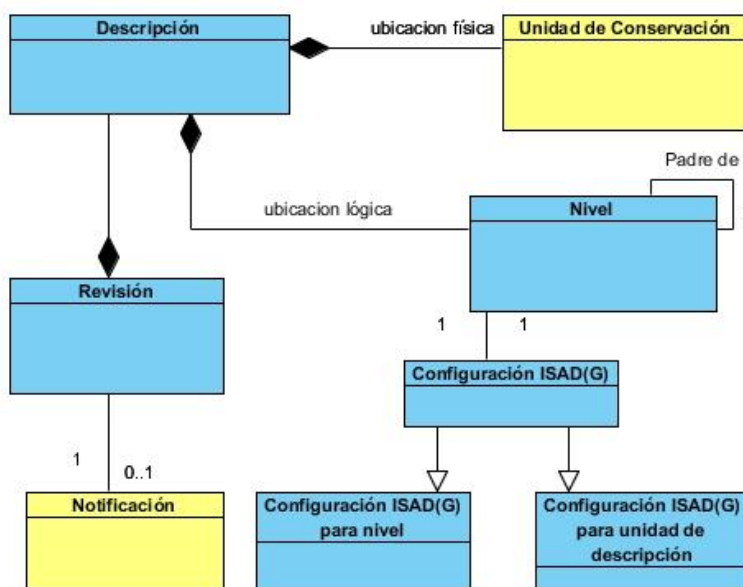


Figura 2.2. Diagrama del Modelo de Dominio. Configuración de la norma ISAD (G).

2.4 Requerimientos

El análisis de requerimientos trata de capturar y describir detalladamente los requerimientos de funcionalidad y de calidad de servicio del producto que se desarrolla (Drake 2008).

En los próximos epígrafes se detallan los requisitos definidos en la investigación.

2.4.1 Requisitos funcionales del sistema

Los Requisitos Funcionales (RF) son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville 2011).

Como resultado del estudio del negocio se identificaron los siguientes requisitos funcionales.

RF 1: Configurar la norma ISAD (G) para los niveles del cuadro de clasificación.

El sistema debe permitir la configuración de la norma ISAD (G) de los tipos de niveles de un cuadro de clasificación, permitiendo la selección de los campos que serán utilizados en la descripción.

RF 2: Registrar la descripción de los niveles de un cuadro de clasificación

El sistema debe permitir la descripción de los niveles de un cuadro de clasificación, atendiendo a los campos de la norma ISAD (G) que fueron establecidos para los mismos.

RF 3: Ver detalles de las descripciones de los niveles de un cuadro de clasificación

El sistema debe permitir mostrar los datos descritos de un nivel de un cuadro de clasificación, utilizando la configuración de la norma ISAD (G) establecida para el mismo y además facilitar la navegación hacia las descripciones de sus ancestros (según su ubicación lógica) y sus documentos relacionados.

RF 4: Revisar las descripciones de unidades de descripción simples y compuestas

El sistema debe permitir revisar la descripción de una unidad de descripción simple o compuesta, utilizando la configuración de la norma ISAD (G) establecida para la misma.

RF 5: Rectificar las descripciones desaprobadas de unidades de descripción simples y compuestas

El sistema debe permitir rectificar la descripción desaprobada de una unidad de descripción simple o compuesta, utilizando la configuración de la norma ISAD (G) establecida para la misma.

RF 6: Modificar las descripciones aprobadas de unidades de descripción simples y compuestas

El sistema debe permitir modificar la descripción aprobada de una unidad de descripción simple o compuesta, utilizando la configuración de la norma ISAD (G) establecida para la misma.

2.4.2 Requisitos no funcionales del sistema

Especifican criterios que plasman las características operacionales del sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. A menudo se aplican al sistema en su totalidad (Sommerville 2011).

En la propuesta de solución se aplican los requisitos no funcionales definidos por el equipo de arquitectura para el sistema XABAL Arkheia 3.0.

Usabilidad

RnF 1: Ambiente

Para el correcto despliegue del sistema XABAL Arkheia se requiere de un servidor de aplicación dónde también se puede alojar la base de datos. A continuación se especifican sus características.

Hardware (Servidor de aplicación y datos)	
CPU	4 x 2.33 GHz (Intel Xeon 5140 Core2 2.33 GHz)
RAM	4 Gb
HDD	250 Gb (en dependencia del monto de información que la institución desee almacenar)
LAN	1 Gbit
Fuentes de Alimentación	1 x 800W

Tabla 2.2. Especificaciones. Servidor de aplicación.

Para el usuario final se requiere de un puesto de trabajo con acceso a la red de la institución y con un navegador web instalado (Mozilla Firefox 10.0 o superior).

Confiabilidad

RnF 2: Conexión entre servidores

Cuando la conexión entre el servidor de aplicación y el de base de datos se interrumpa, la aplicación mostrará una interfaz de información al usuario de la imposibilidad de realizar cualquier transacción hasta que se restablezca la comunicación.

RnF 3: Tiempo de inactividad

Cuando el tiempo de inactividad de sesión caduque el sistema cerrará la sesión del usuario conectado automáticamente. Dicho tiempo de inactividad podrá ser configurado en el sistema.

Soporte

RnF 4: Estándar de codificación

Se utilizará el estándar de codificación para JAVA, Basado en Convenciones de Código para el lenguaje de programación JAVA™ por Scott Hommel Sun Microsystems Inc.

Interfaz

RnF 5: Proporción

La aplicación debe mostrarse en el monitor al 100 %, mostrándose en todo momento todas las áreas de la interfaz de usuario (encabezado, menú lateral, pie de página, área de trabajo)

Requisitos legales, de derecho de autor y otros.

RnF 6: Distribución

No se permite la distribución de este *software* una vez instalada en la institución cliente.

RnF 7: Modificación

La institución es libre de modificar el *software* a su conveniencia, pero el producto de la modificación no podrá distribuirse.

Estándares aplicables

RnF 8: Descripción de los documentos del tipo transcripción

La descripción de los documentos de archivos se realizará usando la norma de descripción archivística ISAD (G).

RnF 9: Descripción de los documentos del tipo audiovisual

La descripción de los documentos audiovisuales se realizará usando la Guía de Descripción y Valoración de Documentos Audiovisuales.

2.4.3 Técnicas de validación de requisitos

El proceso de validación de requisitos tiene por finalidad comprobar que los requisitos del *software* poseen todos los atributos de calidad: son consistentes, completos, precisos, realistas, verificables y definen lo que el usuario desea del producto final. La realización de estas actividades en este momento pretende evitar los altos costos que significaría el tener que corregir una vez avanzado el desarrollo (“Manual de validación. Validación de requisitos.” 2004). Esta validación puede llevarse a cabo empleando diferentes técnicas, para el desarrollo de la propuesta de solución se utilizó la técnica de Prototipado.

Prototipado

Consiste en construir una maqueta del futuro *software* a partir de los requisitos recogidos en la especificación. Esta maqueta será evaluada por el cliente y usuarios para comprobar su corrección y completitud (“Manual de validación. Validación de requisitos.” 2004). Esta técnica permitió que el cliente (Líder del proyecto XABAL Arkheia) pudiera evaluar la forma en que sus necesidades eran satisfechas por la propuesta elaborada.

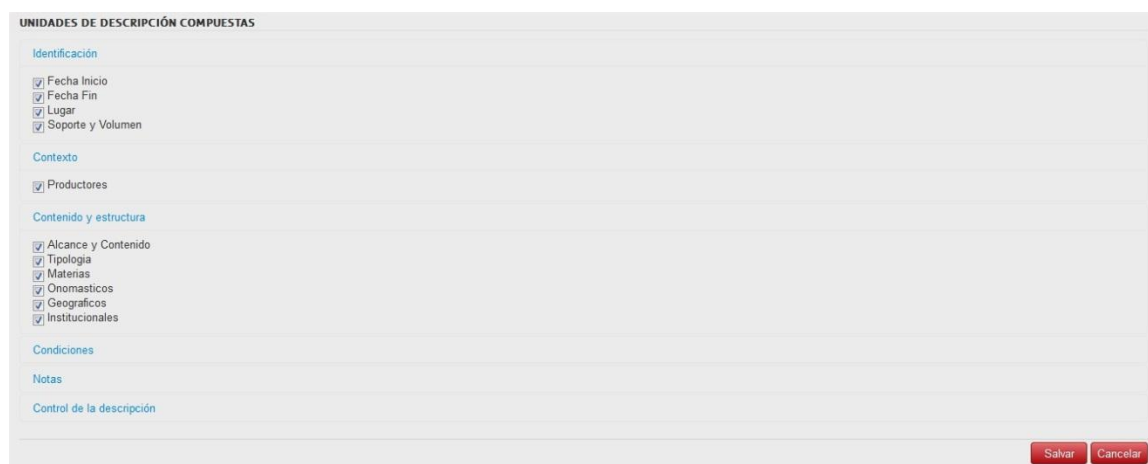


Figura 2.3. Prototipo de Interfaz. Configuración de la norma ISAD (G).

2.5 Modelo de Casos de Uso del Sistema

Constituyen un método alternativo y complementario a los diagramas de contexto para formular los requisitos del sistema. Un caso de uso describe una interacción entre el sistema y un agente externo que se denomina actor (Drake 2008).

2.5.1 Actores

Un actor es una agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema de la misma forma. Es importante tener clara la diferencia entre usuario y actor. Un actor es una clase de rol, mientras que un usuario es una persona que, cuando usa el sistema, asume un rol (Ceria 2005). Los actores son externos al sistema por tanto, al identificar actores se delimita el sistema, y se define su alcance. A continuación, se relacionan los actores del sistema.

Actor	Objetivo
Administrador	Es el encargado de realizar las configuraciones generales del sistema. Entre ellas la configuración de la norma ISAD (G).
Archivero	Es el encargado de procesar los documentos de archivo, registra y modifica sus descripciones.
Revisor	Tiene la responsabilidad de aprobar o desaprobar las descripciones.
Usuario	Sus permisos determinan las acciones que puede realizar en el sistema.

Tabla 2.3. Actores del sistema.

2.5.2 Patrones de Casos de Uso

Técnicas que permiten agrupar los casos de uso de forma que el modelo sea mantenible, reusable y entendible. De forma general, capturan las mejores prácticas para modelar casos de uso. Los patrones empleados fueron:

Múltiples actores - Roles comunes: se emplea cuando a un mismo CU pueden acceder dos o más actores y estos comparten el mismo rol. La aplicación del patrón se puede ver a través del CU Gestionar Audiovisual, que pueden acceder los actores Procesador de Audiovisuales y Revisor de audiovisuales para realizar las mismas acciones, por lo que se establece una relación de herencia entre los actores (Larman 2003).

Extensión-Inclusión: se aplica cuando un flujo puede extender el flujo de otro CU así como ser realizado en sí mismo. Otro caso de aplicación es cuando la extensión del flujo es obligatoria, y se dice que es una inclusión concreta (Larman 2003).

2.5.3 Diagrama de Casos de Uso del Sistema

Captura una vista general de la funcionalidad del sistema con un método muy adecuado para ser interpretado por personas no técnicas como son los usuarios y los expertos de dominio. Está compuesto por actores, casos de uso y sus relaciones.

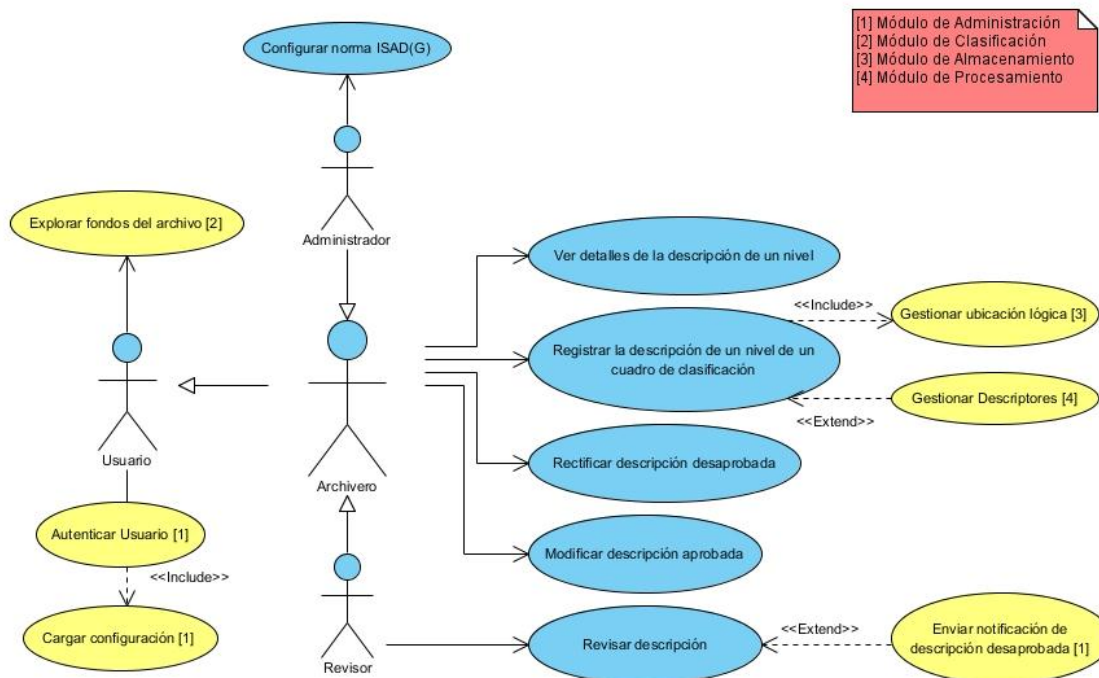


Figura 2.4. Diagrama de casos de uso del sistema.

2.5.4 Especificaciones Casos de Uso

Las Especificaciones de Casos de Uso (ECU) permiten una mejor comprensión y constituyen una descripción absoluta de los requisitos que contienen. Además, facilitan la creación de los Diagramas de Clases del Diseño (DCD).

CU 1. Configurar Norma ISAD (G)

Objetivo	Realizar la configuración de la norma ISAD (G) ya sea para un tipo de nivel de un cuadro de clasificación, para las unidades de descripción simples o las unidades de descripción compuestas.
Actores	Administrador: (Inicia) Configura los campos de la norma ISAD (G) a emplearse en futuras descripciones.
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción “Norma ISAD (G)” en el menú de administración. El sistema muestra la interfaz que permite elegir el tipo de nivel de descripción a configurar. El administrador realiza la configuración y guarda los cambios.
Complejidad	Media
Prioridad	Alta
Precondiciones	Para la configuración de los tipos de niveles de un cuadro de clasificación resulta necesario contar con al menos un cuadro de clasificación.
Postcondiciones	Queda registrada la configuración de la norma ISAD (G) para un nivel de descripción.

Flujo de eventos		
Flujo básico Configurar Norma ISAD (G).		
	Actor	Sistema
1.	En el menú de administración, selecciona la opción "Norma ISAD (G)".	
2.		Muestra una interfaz que permite elegir el tipo de nivel de descripción a configurar. Podrá elegir entre: Configurar la norma ISAD (G) para las unidades de descripción. Configurar la norma ISAD (G) para los tipos de nivel de un cuadro de clasificación específico. Ver Sección 1.
3.	Elige entre configurar la norma para las unidades de descripción simples o las unidades de descripción compuestas.	
4.		Carga la configuración de la norma ISAD (G) para el nivel de descripción seleccionado.
5.	Selecciona los campos de la norma a utilizarse.	
6.	Selecciona la opción guardar.	
7.		Almacena los datos en la BD y muestra un mensaje indicando que los datos se han guardado correctamente.
8.		Regresa a la página que le dio origen.
9.		Termina el caso de uso.
Flujos alternos		
5ª Opción "Cancelar"		
	Actor	Sistema
2.	Selecciona la opción cancelar.	
3.		Retorna a la página que le dio origen.
Sección 1		
	Actor	Sistema
3.	Selecciona el cuadro de clasificación a configurar.	

4.		Muestra una interfaz que permite elegir el tipo de nivel de descripción del cuadro de clasificación seleccionado a configurar.
5.	Selecciona el tipo de nivel de descripción.	
6.		Carga la configuración de la norma ISAD (G) para el tipo de nivel de descripción seleccionado.
7.	Selecciona los campos de la norma a utilizarse.	
8.	Selecciona la opción guardar.	
9.		Almacena los datos en la BD y muestra un mensaje indicando que los datos se han guardado correctamente.
10.		Regresa a la página que le dio origen.
Flujos alternos		
7*a Opción "Cancelar"		
	Actor	Sistema
2.	Selecciona la opción cancelar.	
3.		Retorna a la página que le dio origen.
Relaciones	CU incluidos	No aplica.
	CU extendidos	No aplica.
Requisitos funcionales	no	No aplica.
Asuntos pendientes		No aplica.

Tabla 2.4. ECU CU 1. Configurar Norma ISAD (G).

CU 2. Registrar la descripción de un nivel de un cuadro de clasificación

Objetivo	Realizar descripción de un nivel de un cuadro de clasificación, que podría ser una unidad de descripción simple o compuesta, siguiendo los campos de la norma ISAD (G).
Actores	Archivero: (Inicia) Describe documentos de archivo usando la norma ISAD (G).
Resumen	El caso de uso se inicia cuando el archivero selecciona la opción "Describir nivel" del cuadro de clasificación. El sistema muestra la

	interfaz que permite realizar la descripción mediante la norma ISAD (G). El archivero realiza la descripción y guarda los datos.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	Debe existir una unidad de conservación creada. Debe existir un cuadro de clasificación.	
Postcondiciones	Queda registrada la descripción del nivel.	
Flujo de eventos		
Flujo básico Registrar la descripción de un nivel de un cuadro de clasificación.		
	Actor	Sistema
1.	Selecciona la opción del menú Describir nivel.	
2.		Carga la configuración del tipo de nivel que se desea describir. En caso de tratarse de un documento de archivo ver Sección 1 .
3.		Carga la configuración de la norma ISAD (G) para el nivel de descripción seleccionado.
4.		Muestra un formulario solicitando la información para registrar la descripción de la unidad de descripción según la configuración establecida. El "Título" de la descripción será de llenado obligatorio. De estar habilitados en la configuración los siguientes campos, serán de llenado obligatorio. Área de Identificación <ul style="list-style-type: none"> • Fecha inicial • Fecha final Área de Contenido y Estructura <ul style="list-style-type: none"> • Alcance y contenido Muestra las opciones Guardar, Aceptar y Cancelar.
5.	Introduce los datos solicitados.	

6.	Selecciona la opción "Aceptar".	
7.		Comprueba que se han introducido los datos obligatorios.
8.		Comprueba que los datos introducidos sean correctos.
9.		Almacena los datos en la BD y muestra un mensaje indicando que los datos se han guardado correctamente.
10.		Regresa a la página que le dio origen.
11.		Termina el caso de uso.
Flujos alternos		
5ª Opción "Cancelar"		
	Actor	Sistema
2.	Selecciona la opción cancelar.	
3.		Retorna a la página que le dio origen.
7a Faltan datos obligatorios		
	Actor	Sistema
11		Muestra un mensaje indicando que existen campos obligatorios que no se han completado, señalando los campos obligatorios vacíos.
12		Regresa al paso 7 del flujo básico.
8a Existen datos incorrectos.		
	Actor	Sistema
11		Muestra un mensaje indicando que existen campos con datos incorrectos, señalando los campos.
12		Regresa al paso 7 del flujo básico.
Sección 1		
	Actor	Sistema
3.		Verifica que exista una unidad de conservación creada.
4.		Muestra una interfaz que permite seleccionar el nivel de descripción del documento a describir, sea una unidad

		de descripción simple o compuesta.
5.	Selecciona el nivel de descripción del documento.	
6.		Carga la configuración de la norma ISAD (G) para el nivel de descripción seleccionado.
7.		Muestra una interfaz que permite seleccionar el tipo de documento que desea describir atendiendo a los tipos de cuadro de clasificación disponibles.
8.	Selecciona el tipo de documento a describir.	
9.		<p>Muestra un formulario solicitando la información para registrar la descripción de la unidad de descripción según la configuración establecida.</p> <p>Tanto el “Título”, cómo la “Ubicación Lógica” y la “Unidad de conservación” de la descripción serán de llenado obligatorio.</p> <p>De estar habilitados en la configuración los siguientes campos serán de llenado obligatorio.</p> <p>Área de Identificación</p> <ul style="list-style-type: none"> • Fecha inicial • Fecha final <p>Área de contexto</p> <ul style="list-style-type: none"> • Productores <p>Área de Contenido y Estructura</p> <ul style="list-style-type: none"> • Alcance y contenido <p>Muestra las opciones Guardar, Aceptar y Cancelar.</p>
10.	Introduce los datos solicitados.	
11.	Selecciona la opción “Aceptar”.	
12.		Comprueba que se han introducido los datos obligatorios.
13.		Comprueba que los datos introducidos sean correctos.

14.		Almacena los datos en la BD y muestra un mensaje indicando que los datos se han guardado correctamente.
15.		Regresa a la página que le dio origen.
16.		Termina el caso de uso.
Flujos alternos		
10*a Opción "Cancelar"		
	Actor	Sistema
2.	Selecciona la opción cancelar.	
3.		Retorna a la página que le dio origen.
12a Faltan datos obligatorios		
	Actor	Sistema
11 .		Muestra un mensaje indicando que existen campos obligatorios que no se han completado, señalando los campos obligatorios vacíos.
12 .		Regresa al paso 7 del flujo básico.
13a Existen datos incorrectos.		
	Actor	Sistema
11 .		Muestra un mensaje indicando que existen campos con datos incorrectos, señalando los campos.
12 .		Regresa al paso 7 del flujo básico.
Relaciones	CU incluidos	<u>Gestionar ubicación lógica</u>
	CU extendidos	<u>Registrar descriptores</u> en el CU <u>Registrar Descriptores.</u>
Requisitos funcionales	no	No aplica.
Asuntos pendientes		No aplica.

Tabla 2.5. ECU CU 2. Registrar la descripción de un nivel de un cuadro de clasificación.

CU 3. Ver detalles de la descripción de un nivel

Objetivo	Ver la descripción realizada para un nivel de un cuadro de clasificación. Ver los documentos relacionados a dicho nivel y sus descripciones correspondientes.	
Actores	Archivero: (Inicia) Ve la descripción realizada para un nivel de un cuadro de clasificación	
Resumen	El caso de uso se inicia cuando el archivero selecciona la opción “Ver descripción”. El sistema muestra la interfaz con la descripción realizada con anterioridad y permite la navegación hacia las descripciones de los documentos relacionados al nivel.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	Se debe haber descrito el nivel.	
Postcondiciones		
Flujo de eventos		
Flujo básico Ver detalles de la descripción de un nivel.		
	Actor	Sistema
1.	Selecciona la opción “Ver descripción”	
2.		Muestra una interfaz que, teniendo en cuenta la configuración de la norma ISAD (G) para el nivel de descripción en cuestión, muestra los datos descritos del nivel. Además permite la navegación hacia los niveles ancestros así como los documentos relacionados al nivel.
3.	Selecciona la opción “Aceptar”	
4.		Regresa a la página que le dio origen.
5.		Termina el caso de uso.
Flujos alternos		
3a Opción “Ver ubicación lógica”		
	Actor	Sistema
4.	Selecciona la opción “Ver ubicación lógica”.	
5.		Muestra la rama del cuadro de clasificación que lleva al nivel en cuestión y permite acceder a la descripción de los niveles ancestros.
6.	Selecciona la opción “Aceptar”	
7.		Retorna al paso 2.

Relaciones	CU incluidos	No aplica.
	CU extendidos	No aplica.
Requisitos funcionales no	No aplica.	
Asuntos pendientes	No aplica.	

Tabla 2.6. ECU CU 3. Ver detalles de la descripción de un nivel.

CU 4. Revisar descripción

Objetivo	Revisar la descripción de un nivel de un cuadro de clasificación.	
Actores	Revisor: (Inicia) Revisa la descripción de un nivel de un cuadro de clasificación.	
Resumen	El caso de uso se inicia cuando el revisor selecciona la opción “Revisar descripción” en el menú de “Procesamiento”. El sistema muestra las posibles descripciones a revisar. El revisor aprueba o desaprueba la descripción.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	Debe existir al menos una descripción registrada.	
Postcondiciones	Queda aprobada o desaprobada la descripción del nivel.	
Flujo de eventos		
Flujo básico Revisar Descripción.		
	Actor	Sistema
1.	Selecciona la opción del menú “Revisar descripción”	
2.		Muestra las posibles unidades de conservación a revisar.
3.	Selecciona la unidad de conservación del documento a revisar.	
4.		Muestra las descripciones a revisar en la unidad de conservación seleccionada.
5.	Selecciona la descripción a revisar.	
6.		Muestra un formulario con la información descrita, según la configuración de la norma ISAD (G) establecida para el nivel

		<p>de descripción del documento.</p> <p>Tanto el “Título”, cómo la “Ubicación Lógica” y la “Unidad de conservación” de la descripción serán de llenado obligatorio.</p> <p>De estar habilitados en la configuración los siguientes campos serán de llenado obligatorio.</p> <p>Área de Identificación</p> <ul style="list-style-type: none"> • Fecha inicial • Fecha final <p>Área de contexto</p> <ul style="list-style-type: none"> • Productores <p>Área de Contenido y Estructura</p> <ul style="list-style-type: none"> • Alcance y contenido <p>Se tiene además las opciones de aprobar o desaprobar la descripción.</p> <p>Muestra las opciones Guardar, Aceptar y Cancelar.</p>
7.	Aprueba o desaprueba la descripción y selecciona la opción aceptar.	
8.		Comprueba que se han introducido los datos obligatorios.
9.		Comprueba que los datos introducidos sean correctos.
10.		<p>Almacena los datos en la BD y muestra un mensaje indicando que los datos se han guardado correctamente.</p> <p>En el caso de haberse desaprobadado la descripción, notifica al archivero encargado de la misma.</p>
11.		Regresa a la página que le dio origen.
12.		Termina el caso de uso.
Flujos alternos		
*a Opción “Cancelar”		

	Actor	Sistema
2.	Selecciona la opción cancelar.	
3.		Retorna a la página que le dio origen.
1ª Faltan datos obligatorios		
	Actor	Sistema
11		Muestra un mensaje indicando que existen campos obligatorios que no se han completado, señalando los campos obligatorios vacíos.
12		Regresa al paso 6 del flujo básico.
2ª Existen datos incorrectos.		
	Actor	Sistema
11		Muestra un mensaje indicando que existen campos con datos incorrectos, señalando los campos.
12		Regresa al paso 6 del flujo básico.
Relaciones	CU incluidos	
	CU extendidos	<u>Registrar descriptores</u> en el CU <u>Registrar Descriptores. Enviar notificación de descripción desaprobada.</u>
Requisitos funcionales	no	No aplica.
Asuntos pendientes		No aplica.

Tabla 2.7. ECU CU 4. Revisar descripción.

CU 5. Rectificar descripción desaprobada

Objetivo	Rectificar la descripción de un documento de archivo desaprobada con anterioridad.
Actores	Archivero: (Inicia) Modifica la descripción de un documento de archivo desaprobada con anterioridad.
Resumen	El caso de uso se inicia cuando el archivero selecciona la opción “Rectificar desaprobadas” del menú de “Procesamiento”. El sistema muestra la interfaz que permite realizar la descripción mediante la norma ISAD (G). El archivero realiza las modificaciones a la descripción y

	guarda los datos.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	Debe existir una descripción de documento desaprobada.	
Postcondiciones	Queda modificada la descripción de documento.	
Flujo de eventos		
Flujo básico Modificar descripción.		
	Actor	Sistema
1.	Selecciona la opción "Rectificar desaprobadas".	
2.		Muestra una interfaz con la lista de las descripciones desaprobadas y opciones de filtrado que permiten al archivero acotar la búsqueda de la descripción a modificar atendiendo a su ubicación física o a las fechas en que se realizó la revisión.
3.	Selecciona la descripción a modificar	
4.		<p>Muestra un formulario con la información descrita, según la configuración de la norma ISAD (G) establecida para el nivel de descripción del documento.</p> <p>Tanto el "Título", cómo la "Ubicación Lógica" y la "Unidad de conservación" de la descripción serán de llenado obligatorio.</p> <p>De estar habilitados en la configuración los siguientes campos serán de llenado obligatorio.</p> <p>Área de Identificación</p> <ul style="list-style-type: none"> • Fecha inicial • Fecha final <p>Área de contexto</p> <ul style="list-style-type: none"> • Productores <p>Área de Contenido y Estructura</p> <ul style="list-style-type: none"> • Alcance y contenido

5.	Modifica los datos deseados y selecciona la opción "Aceptar".	
6.		Comprueba que se han introducido los datos obligatorios.
7.		Comprueba que los datos introducidos sean correctos.
8.		Almacena los datos en la BD y muestra un mensaje indicando que los datos se han guardado correctamente.
9.		Regresa a la página que le dio origen.
10		Termina el caso de uso.
Flujos alternos		
*a Opción "Cancelar"		
	Actor	Sistema
3.	Selecciona la opción cancelar.	
4.		Retorna a la página que le dio origen.
1a Filtra la búsqueda		
	Actor	Sistema
3.	Entra los parámetros por los que se quiere filtrar la búsqueda. Pueden ser la unidad de conservación o las fechas en que se realizó la revisión.	
4.		Acota la lista de posibles descripciones a modificar atendiendo a los parámetros de filtrado.
5.		Regresa al paso 2 del flujo básico.
2a Faltan datos obligatorios		
	Actor	Sistema
7.		Muestra un mensaje indicando que existen campos obligatorios que no se han completado, señalando los campos obligatorios vacíos.
8.		Regresa al paso 4 del flujo básico.
3a Existen datos incorrectos.		
	Actor	Sistema
7.		Muestra un mensaje indicando que

		existen campos con datos incorrectos, señalando los campos.
8.		Regresa al paso 4 del flujo básico.
Relaciones	CU incluidos	<u>Gestionar ubicación lógica</u>
	CU extendidos	<u>Registrar descriptores</u> en el CU <u>Registrar Descriptores.</u>
Requisitos funcionales	no	No aplica.
Asuntos pendientes		No aplica.

Tabla 2.8. ECU CU 5. Rectificar descripción desaprobada.

CU 6. Modificar descripción aprobada

Objetivo	Modificar la descripción de un documento de archivo aprobada con anterioridad.	
Actores	Archivero: (Inicia) Modifica la descripción de un documento de archivo aprobada con anterioridad.	
Resumen	El caso de uso se inicia cuando el archivero selecciona la opción “Modificar aprobadas” del menú de “Procesamiento”. El sistema muestra la interfaz que permite realizar la descripción mediante la norma ISAD (G). El archivero realiza las modificaciones a la descripción y guarda los datos.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	Debe existir una descripción de documento aprobada.	
Postcondiciones	Queda modificada la descripción de documento.	
Flujo de eventos		
Flujo básico Modificar descripción.		
	Actor	Sistema
1.	Selecciona la opción “Modificar aprobadas”.	
2.		Muestra una interfaz con la lista de las descripciones aprobadas y opciones de filtrado que permiten al archivero acotar la búsqueda de la descripción a modificar atendiendo a su ubicación física o a las fechas en que se realizó la revisión.

3.	Selecciona la descripción a modificar	
4.		<p>Muestra un formulario con la información descrita, según la configuración de la norma ISAD (G) establecida para el nivel de descripción del documento.</p> <p>Tanto el “Título”, cómo la “Ubicación Lógica” y la “Unidad de conservación” de la descripción serán de llenado obligatorio.</p> <p>De estar habilitados en la configuración los siguientes campos serán de llenado obligatorio.</p> <p>Área de Identificación</p> <ul style="list-style-type: none"> • Fecha inicial • Fecha final <p>Área de contexto</p> <ul style="list-style-type: none"> • Productores <p>Área de Contenido y Estructura</p> <ul style="list-style-type: none"> • Alcance y contenido
5.	Modifica los datos deseados y selecciona la opción “Aceptar”.	
6.		Comprueba que se han introducido los datos obligatorios.
7.		Comprueba que los datos introducidos sean correctos.
8.		Almacena los datos en la BD y muestra un mensaje indicando que los datos se han guardado correctamente.
9.		Regresa a la página que le dio origen.
10		Termina el caso de uso.
Flujos alternos		
*a Opción “Cancelar”		
	Actor	Sistema
3.	Selecciona la opción cancelar.	
4.		Retorna a la página que le dio origen.

1a Filtra la búsqueda		
	Actor	Sistema
3.	Entra los parámetros por los que se quiere filtrar la búsqueda. Pueden ser la unidad de conservación o las fechas en que se realizó la revisión.	
4.		Acota la lista de posibles descripciones a modificar atendiendo a los parámetros de filtrado.
5.		Regresa al paso 2 del flujo básico.
2a Faltan datos obligatorios		
	Actor	Sistema
7.		Muestra un mensaje indicando que existen campos obligatorios que no se han completado, señalando los campos obligatorios vacíos.
8.		Regresa al paso 4 del flujo básico.
3a Existen datos incorrectos.		
	Actor	Sistema
7.		Muestra un mensaje indicando que existen campos con datos incorrectos, señalando los campos.
8.		Regresa al paso 4 del flujo básico.
Relaciones	CU incluidos	<u>Gestionar ubicación lógica</u>
	CU extendidos	<u>Registrar descriptores</u> en el CU <u>Registrar Descriptores.</u>
Requisitos funcionales	no	No aplica.
Asuntos pendientes		No aplica.

Tabla 2.9. ECU CU 6. Modificar descripción aprobada.

La tabla 2.11 contiene la relación entre los casos de uso identificados y los requisitos contenidos en ellos.

RF(15) / CU(6)	CU 1	CU 2	CU 3	CU 4	CU 5	CU 6
RF 1	X					
RF 2		X				
RF 3			X			
RF 4				X		
RF 5					X	
RF 6						X

Tabla 2.10. Relación entre Requisitos funcionales y Casos de Uso.

2.6 Diseño de la solución

A continuación se desarrollan los procesos comprendidos en el diseño del sistema. Entre los que se comprenden la definición de la arquitectura y los patrones de diseño a emplear así como los diagramas de clases del diseño.

2.6.1 Arquitectura del sistema

Vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema (Arciniegas Herrera 2001).

Arquitectura Grails

La propuesta de solución se basa arquitectónicamente en el framework Grails, al igual que el sistema XABAL Arkheia 3.0. Grails es representado como una arquitectura en capas sobre la máquina virtual de Java, programada con los lenguajes de programación Groovy y Java. El siguiente diagrama muestra los principales componentes de la arquitectura Grails.

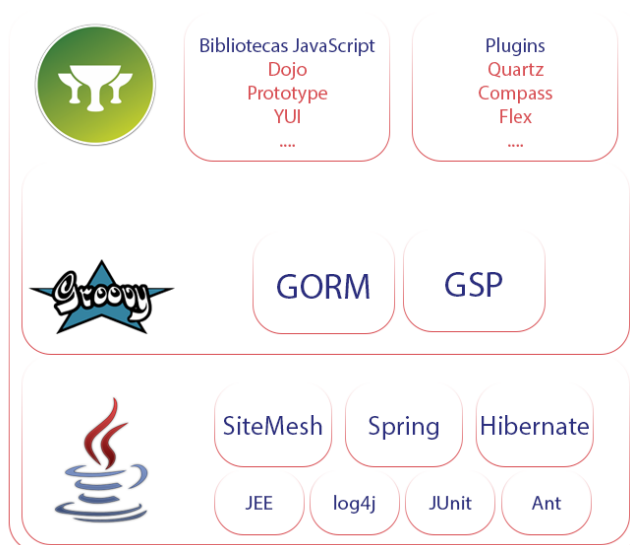


Figura 2.5. Principales componentes de la arquitectura Grails.

Todo sistema desarrollado con este framework sigue el patrón arquitectónico MVC (Modelo Vista Controlador) donde el modelo representa las entidades del problema, las vistas manejan la presentación de los datos al usuario y los controladores proporcionan significado a las peticiones del usuario, actuando sobre los datos representados por el modelo.

La arquitectura de Grails está concebida por 3 capas lógicas principales: *Capa web*, *Capa de servicios* y *Capa de datos*, cada capa está separada de la siguiente, su interacción se lleva a cabo mediante interfaces que definen funcionalidades que la misma debe brindar, también llamadas fachadas cuya función fundamental es asegurar que el acoplamiento sea el más bajo posible y la abstracción del funcionamiento de la capa inferior (Brito Calahorro 2009).

Capa web: *La capa web se compone de tres partes principales: vistas, modelos y controladores, teniendo en Controllers las clases controladoras, en Domain las clases del dominio y en Views las Groovy Server Pages o más conocidas como GSP (Brito Calahorro 2009).*

Capa de servicios: *La capa de servicios se encarga de encapsular toda la lógica de la aplicación en fachadas de negocio que son utilizadas por los controladores en la capa de presentación, dejándoles el manejo de flujo de solicitudes con redirecciones. Sus clases según la arquitectura propuesta por Grails radicarán en el paquete Services (Brito Calahorro 2009).*

Capa de datos: *La capa de datos se encarga de manejar los objetos de acceso a datos separándolos del mecanismo de persistencia utilizado, mediante las interfaces que exponen las operaciones de persistencia. Grails para evitar a los programadores tener que trabajar directamente con el sistema gestor de base de datos y tablas, permite trabajar con objetos en su lugar. Utiliza Hibernate, la biblioteca más popular para Java, como una herramienta de Mapeo Objeto-Relacional (ORM). Sin embargo dada la naturaleza dinámica de Grails y la adopción del convenio sobre la configuración, se crea sobre una versión superior de la implementación de Hibernate llamado Grails Mapeo Objeto-Relacional (GORM) que simplifica el trabajo con Hibernate y elimina cualquier configuración externa (Udd y F. N 2008).*

2.6.2 Patrones de diseño

Un patrón de diseño está relacionado con los aspectos del diseño de los subsistemas. Es una solución estándar para un problema común de programación y una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios (Larman 2003).

Los patrones hacen el diseño orientado a objetos más flexible, elegante y en algunos casos reusable.

A continuación se relacionan y describen los patrones seguidos por el diseño de la solución propuesta.

GRASP

Los patrones generales de *software* para asignación de responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales de la asignación de

responsabilidades a objetos. El nombre se eligió para indicar la importancia de captar estos principios, con el objetivo de diseñar el *software* de manera eficaz (Larman 2003).

Controlador

Un controlador es un objeto responsable del manejo de los eventos del sistema que no pertenece a la interfaz del usuario, el controlador recibe la solicitud del servicio desde la capa de presentación y coordina su realización delegando a otros objetos. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación para aumentar la reutilización de código y a su vez tener un mayor control del flujo de eventos del sistema, se encarga de asignar responsabilidades a clases específicas facilitando la centralización de actividades como la validación, la seguridad, entre otras.

Este patrón está presente en los controladores del sistema. Ejemplo de ello es la clase ConfigISADGController la cual se ocupa de atender las peticiones referentes a la configuración de la norma ISAD (G), tratarlas con el apoyo de la clase de servicios ConfigISADGService y presentar las posibles respuestas al usuario mediante las vistas.

Alta cohesión

Sugiere asignar responsabilidades a las clases procurando que la cohesión sea lo más alta posible.

Se pone de manifiesto cuando una clase tiene responsabilidades moderadas en un área funcional y colabora con otras. Una clase con mucha cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. Su alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y las mejoras (Larman 2003).

Aplicado en la mayoría de las clases de la aplicación, fundamentalmente en las controladoras, y las pertenecientes a la capa de servicios. Ejemplo de ello es la clase ConfigISADGService la cual presenta cohesión entre sus procedimientos, brindando transparencia, facilitando su comprensión así como simplificando su mantenimiento en caso de ser necesario.

Bajo acoplamiento

El acoplamiento indica que tan fuertemente está conectada o influye una clase con otra. Una clase con bajo acoplamiento no depende de otras clases.

Se evidencia cuando una clase referencia a otra en forma de atributo (X tiene una variable de tipo Y), es una subclase directa o indirecta de otra (X hereda de Y), tiene un método que en su implementación referencia una instancia de otra o implementa una interfaz (Y es una interfaz y X la implementa). Las clases con bajo acoplamiento no se afectan por cambios de otros componentes, son fáciles de comprender por separado y fáciles de reutilizar (Larman 2003).

Este patrón es seguido en buena parte de las clases de la aplicación. Ejemplo de ello es la clase DConfigISADGnd, la misma no contiene relaciones a otras clases de la aplicación, de tal forma que no se vería afectada por cambios en cualquier otra clase del sistema.

Experto

Sugiere asignar responsabilidades al experto de la información, es decir, a la clase que tiene la información necesaria para llevar la tarea a cabo.

Conserva el encapsulamiento, debido a que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento (Larman 2003).

En el sistema se delegan las responsabilidades a las clases expertas. Un ejemplo de ello es `ConfigISADGController` la cual maneja toda la información referente a las configuraciones de la norma ISAD (G) en el sistema.

Patrones GoF

Los patrones de la banda de cuatro (GoF por sus siglas en inglés) son llamados así por encontrarse recogidos en el libro *Patrones de Diseño*, publicado en 1994 por cuatro autores: Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.

Singleton

Este patrón consiste en crear una instancia de un objeto, para toda la aplicación, sería como un mecanismo de visibilidad global de dicha instancia. Por defecto, en Grails todos los servicios utilizan este patrón. Solo existe una instancia de la clase que se inyecta en todos los artefactos que declaren la variable correspondiente (Brito Calahorro 2009).

Cuando se ejecuta la aplicación, Grails configura el contenedor Spring para instanciar estas clases una única vez, creando un punto de acceso global a ellas, para que puedan ser utilizadas luego sin necesidad de volver a instanciarse. En la aplicación un ejemplo de este patrón es el servicio `ConfigISADGService`.

Inversión de control (IoC)

La inversión de control es otro patrón utilizado por Grails, a través del cual las dependencias de un componente no deben ser gestionadas desde el propio componente para que este solo contenga la lógica necesaria para hacer su trabajo. El objetivo de esta técnica es mantener los componentes lo más sencillos que sea posible, incluyendo únicamente código que tenga relación con la lógica de negocio, logrando que la aplicación sea más fácil de comprender y mantener (Brito Calahorro 2009).

En Grails esto es posible ya que el contenedor Spring es configurado para que gestione el ciclo de vida de los componentes y sus dependencias.

2.6.3 Diagrama de clases del diseño

A diferencia del Modelo Conceptual, un Diagrama de Clases de Diseño (DCD) muestra representaciones de entidades de *software* más que conceptos del mundo real. En ellos se evidencian: las clases con sus atributos, métodos y asociaciones, las interfaces y la navegabilidad entre ellas así como sus dependencias.

Las figuras 2.5 y 2.6 muestran los Diagramas de Clases de Diseño de los casos de uso más relevantes.

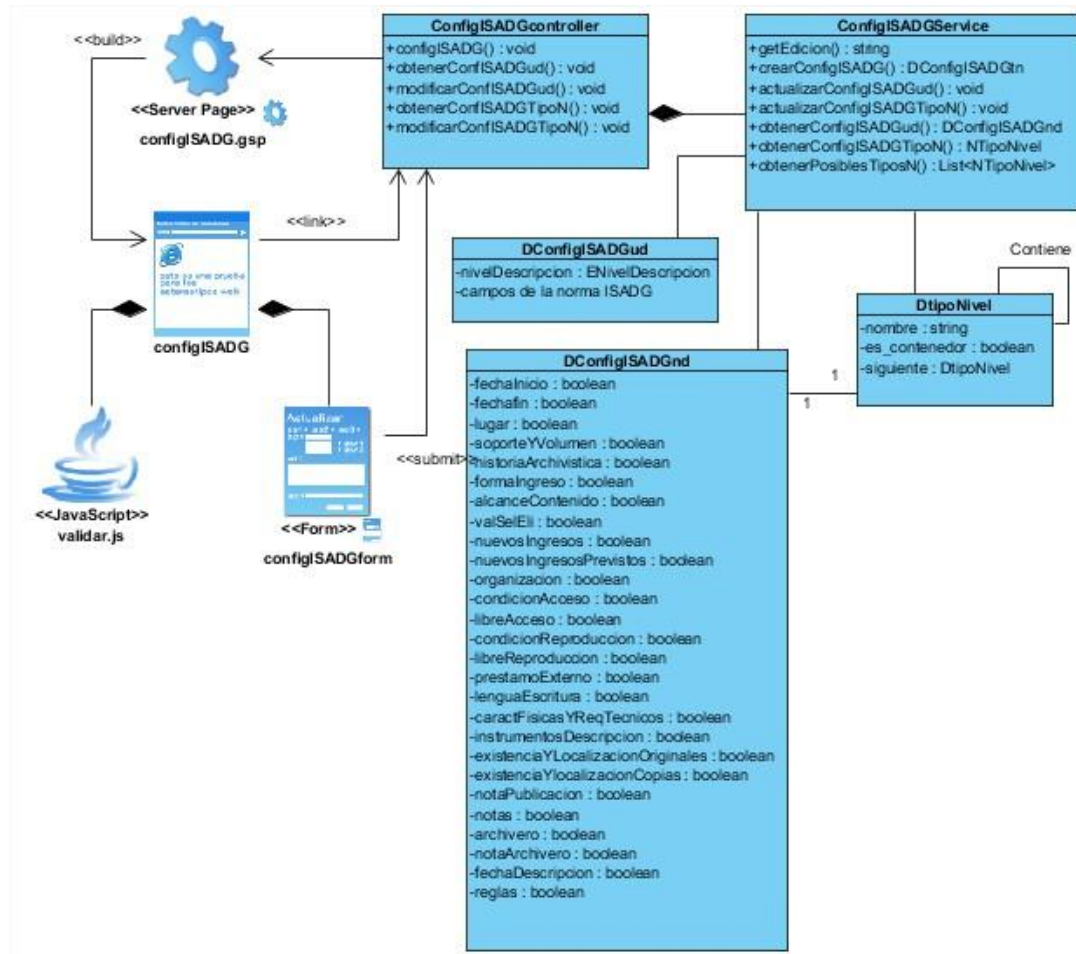


Figura 2.6. DCD. CU Configurar norma ISAD (G).

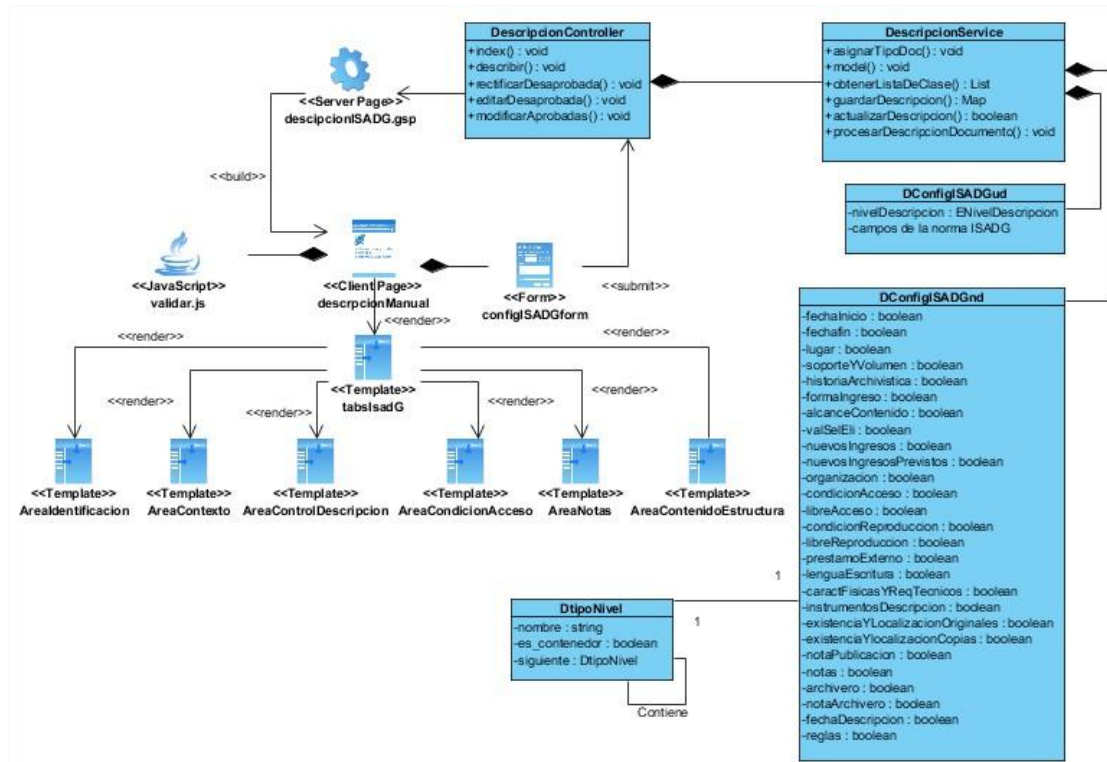


Figura 2.7. DCD. CU Describir nivel.

2.6.4 Validación del diseño propuesto

Con el objetivo de producir un sistema o producto de alta calidad se emplean un conjunto de métricas que permiten ofrecer una valoración cuantitativa de la calidad del software.

Una métrica es un instrumento que cuantifica un criterio y persigue comprender mejor la calidad del producto, estimar la efectividad del proceso y mejorar la calidad del trabajo realizado al nivel del proyecto (Pressman 2009).

Para la evaluación de la calidad del diseño propuesto se utilizaron los siguientes atributos de calidad:

Responsabilidad: consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.

Complejidad de implementación: consiste en el grado de dificultad que implica la implementación de un diseño de clases determinado.

Reutilización: consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.

Acoplamiento: consiste en el grado de dependencia o interconexión de una clase o estructura de clase con otras, está muy ligada a la característica de Reutilización.

Complejidad del mantenimiento: consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costos y la planificación del proyecto.

Cantidad de pruebas: consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad del producto diseñado.

A continuación se describen los instrumentos seleccionados para validar la calidad del diseño y su relación con los atributos de calidad definidos.

Tamaño Operacional de Clase (TOC)

Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:

Atributo de calidad	Modo en que lo afecta un aumento del TOC
Responsabilidad	Aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Aumento de la complejidad de implementación de la clase.
Reutilización	Disminución en el grado de reutilización de la clase.

Tabla 2.11. Relación entre el valor del TOC y los atributos de calidad que evalúa.

La tabla que se muestra a continuación contiene el rango de valores para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización). La variable *Prom* indica el promedio de operaciones por cada clase.

Atributo de calidad	Categoría	Criterio
Responsabilidad	Baja	TOC <= Prom
	Media	Prom < TOC <2*Prom
	Alta	TOC > 2*Prom
Complejidad de implementación	Baja	TOC <= Prom
	Media	Prom < TOC <2*Prom
	Alta	TOC > 2*Prom
Reutilización	Baja	TOC > 2*Prom
	Media	Prom < TOC <2*Prom
	Alta	TOC <= Prom

Tabla 2.12. Rangos de valores para la evaluación técnica del TOC

La siguiente tabla muestra los umbrales de la técnica TOC.

TOC	Criterio
Pequeño	TOC <= Prom
Medio	Prom < TOC <2*Prom
Grande	TOC > 2*Prom

Tabla 2.14. Umbrales para el TOC.

Resultados de la evaluación (TOC):

Representación en por ciento de los resultados obtenidos en el instrumento, agrupados en los intervalos definidos.

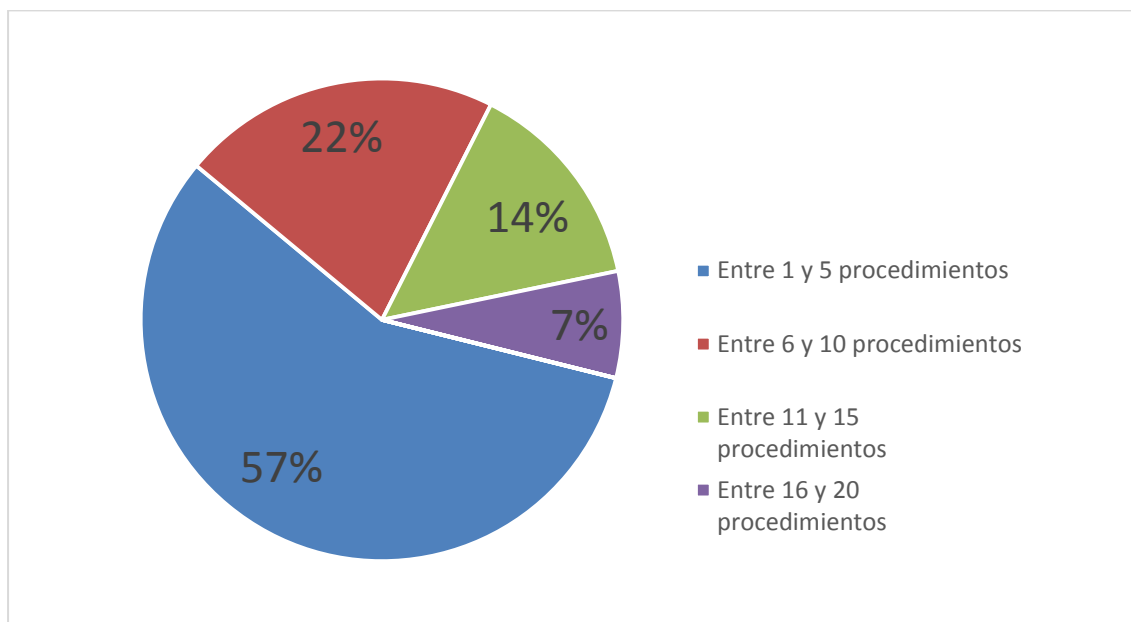


Figura 2.8. Resultados de la métrica TOC.

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.

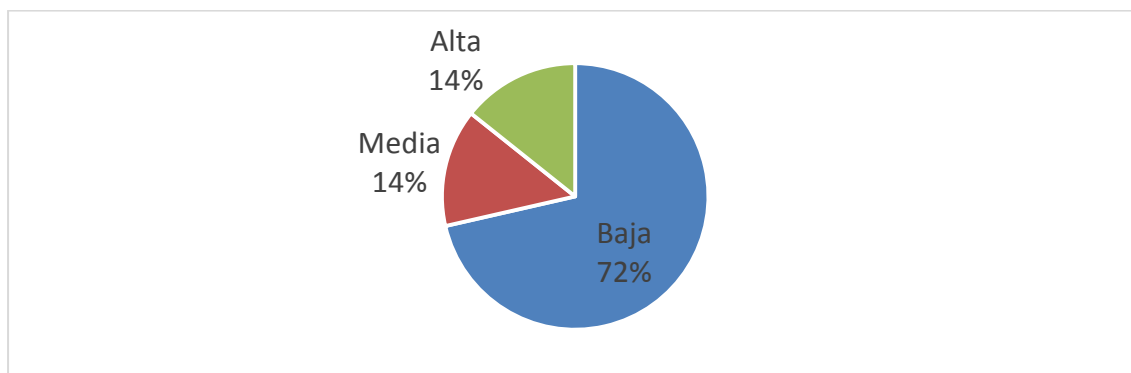


Figura 2.9. Resultados de la métrica TOC. Atributo responsabilidad.

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación.

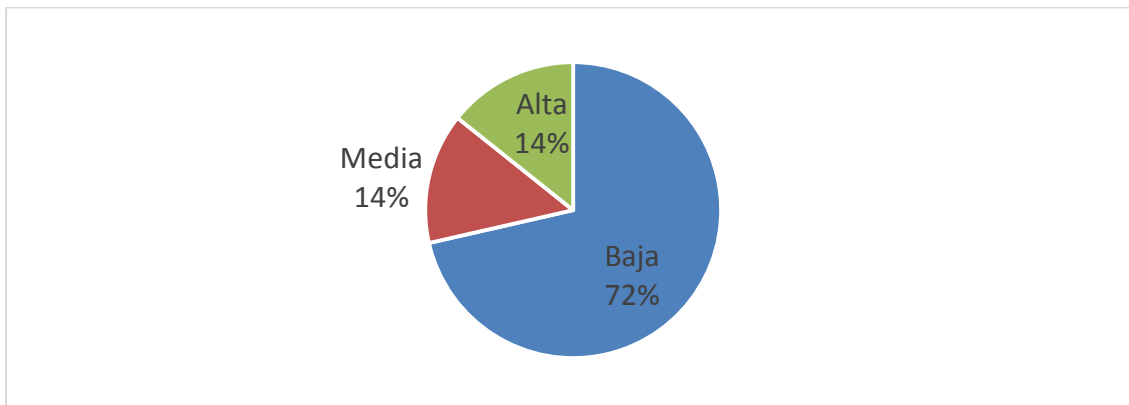


Figura 2.10. Resultados de la métrica TOC. Atributo complejidad de implementación.

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.

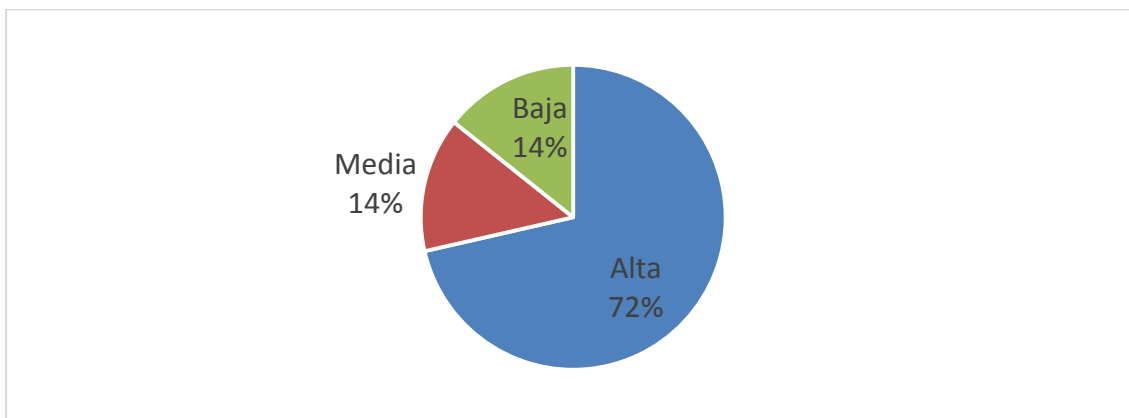


Figura 2.11. Resultados de la métrica TOC. Atributo reutilización.

Luego de analizar los resultados obtenidos con la aplicación de la métrica Tamaño operacional de clase resulta posible concluir que el diseño propuesto posee la calidad requerida, teniendo en consideración que el 72% de las clases presentan calificación “baja” para los atributos de responsabilidad y complejidad de implementación mientras que para el atributo reutilización su valoración es alta. Lográndose de esta manera fomentar la reutilización mientras que se mantienen en niveles mínimos la complejidad de implementación y la responsabilidad de las clases proyectadas en el diseño.

Relaciones entre Clases (RC)

Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad:

Atributo de calidad	Modo en que lo afecta un aumento del RC
Acoplamiento	Aumento del acoplamiento de la clase.
Complejidad del mantenimiento	Aumento de la complejidad del mantenimiento de la clase.
Cantidad de pruebas	Aumento de la cantidad de pruebas unitarias necesarias para probar una clase.
Reutilización	Disminución en el grado de reutilización de la clase.

Tabla 2.14. Relación entre el valor del RC y los atributos de calidad que evalúa.

La tabla que se muestra a continuación contiene el rango de valores para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad del mantenimiento, Cantidad de pruebas y Reutilización). La variable *Prom* indica el promedio de operaciones por cada clase.

Atributo de calidad	Categoría	Criterio
Responsabilidad	Ninguno	0
	Bajo	1
	Medio	2
	Alto	$RC > 2$
Complejidad de implementación	Baja	$RC \leq Prom$
	Media	$Prom < RC < 2 * Prom$
	Alta	$RC > 2 * Prom$
Complejidad de implementación	Baja	$RC \leq Prom$
	Media	$Prom < RC < 2 * Prom$
	Alta	$RC > 2 * Prom$
Reutilización	Baja	$RC > 2 * Prom$
	Media	$Prom < RC < 2 * Prom$
	Alta	$RC \leq Prom$

Tabla 2.15. Rangos de valores para la evaluación técnica del RC.

Resultados de la evaluación (RC):

Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

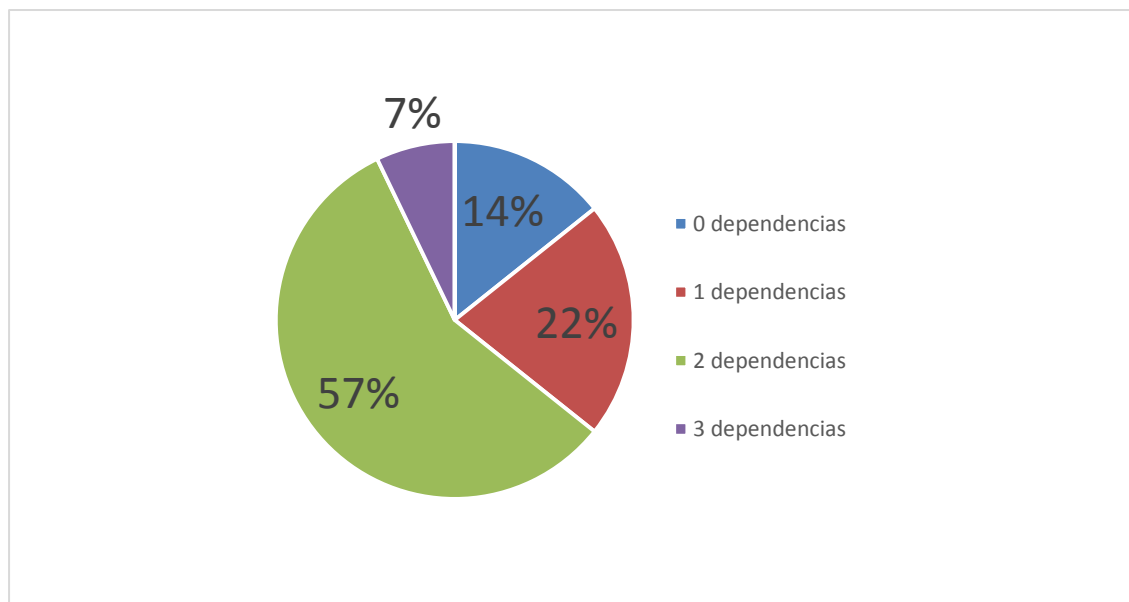


Figura 2.12. Resultados de la métrica RC.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

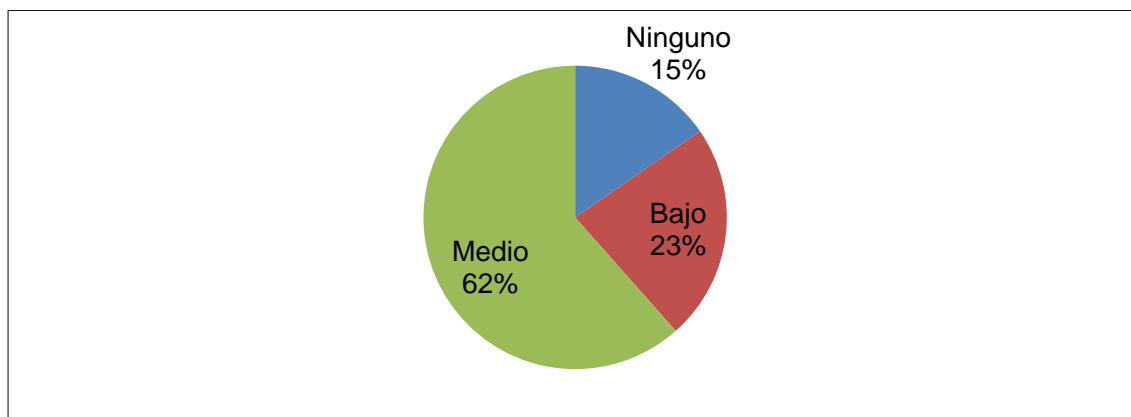


Figura 2.13. Resultados de la métrica RC. Atributo acoplamiento.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento.

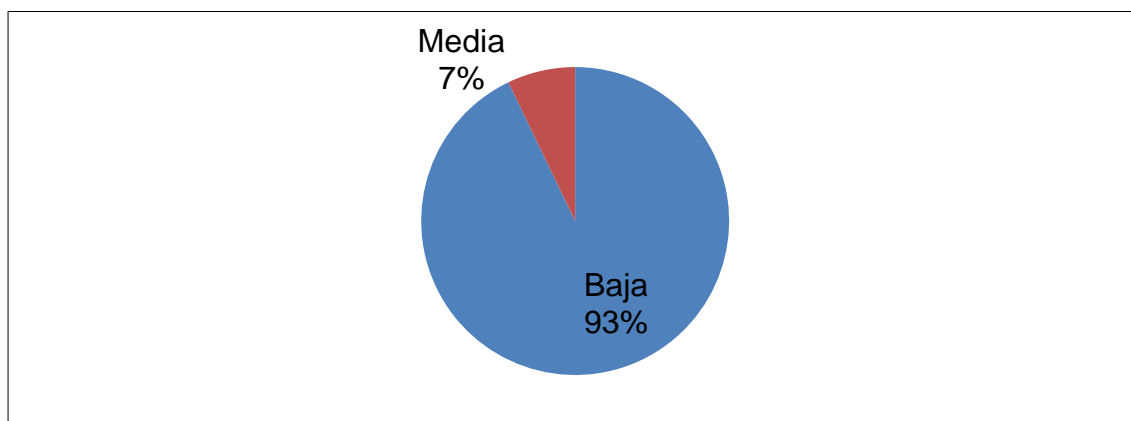


Figura 2.14. Resultados de la métrica RC. Atributo complejidad de mantenimiento.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas.

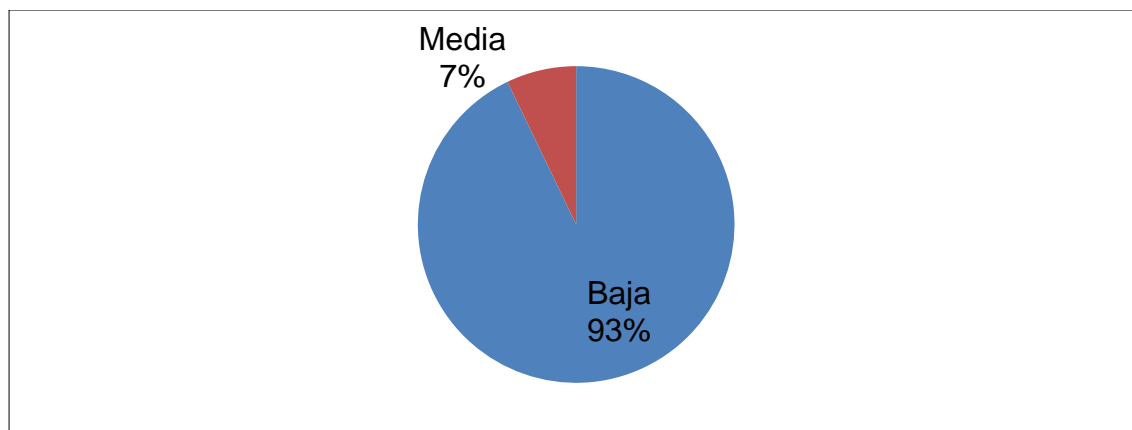


Figura 2.15. Resultados de la métrica RC. Atributo cantidad de pruebas.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.

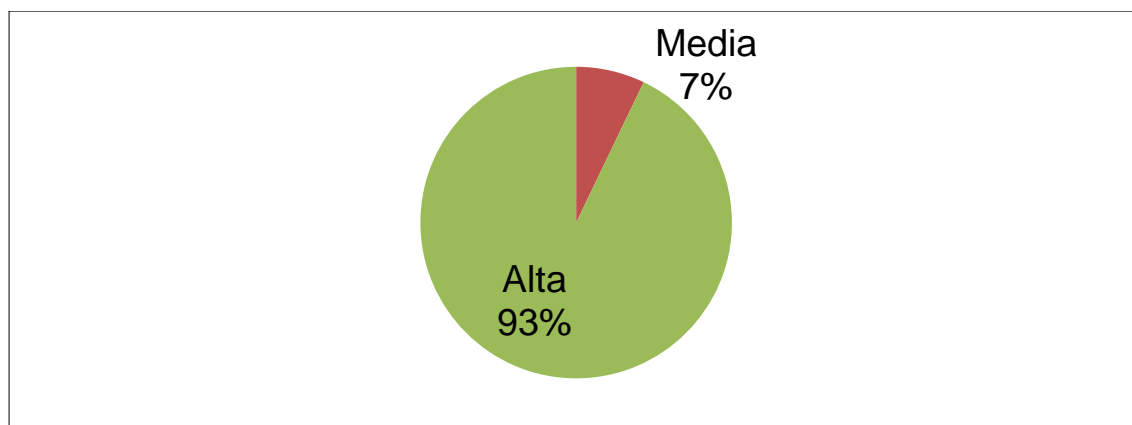


Figura 2.16. Resultados de la métrica RC. Atributo reutilización.

Al igual que la métrica aplicada con anterioridad, los resultados arrojados por la aplicación de la métrica Relaciones entre clases avalan la buena calidad del diseño propuesto. En este caso fueron evaluados los atributos: acoplamiento, el cuál evidencia que la totalidad de las clases poseen 2 o menos dependencias, mientras que la Complejidad de mantenimiento, la Cantidad de pruebas y la Reutilización se comportan favorablemente para el 93% de las clases.

2.7 Conclusiones del Capítulo

A partir de la modelación de los principales conceptos del dominio se logró una mejor comprensión del negocio, identificando las necesidades del cliente que permitieron definir los requerimientos funcionales y no funcionales que debe cumplir el sistema. Con los requerimientos identificados, se elaboró el Modelo de Casos de Uso del Sistema, que apoyado por patrones resultó mantenible, reusable y entendible.

El análisis de la arquitectura propuesta por Grails y la forma en que esta es adoptada por el sistema XABAL Arkheia 3.0 en conjunto con el estudio de los patrones del diseño de *software*, permitieron contar con los elementos necesarios para la elaboración de un diseño de la solución con la calidad requerida, la cual es avalada por la aplicación de métricas como TOC y RC, sirviendo así como referencia para la implementación de la propuesta.

Capítulo 3. Implementación y prueba

3.1 Introducción al capítulo

En el presente capítulo se analizan algunos artefactos necesarios para proceder a la implementación del sistema. Ejemplo de ello son los diagramas de componentes y de despliegue. Además se verifica el cumplimiento de los requerimientos mediante las pruebas de *software*.

3.2 Diagramas de componentes

Los diagramas de componentes son utilizados para modelar los componentes del sistema, incluyen además los artefactos que son implementados por estos componentes y las relaciones entre ellos (Pressman 2009). En ellos se modelarán las librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema. Para una mejor comprensión, dichos componentes fueron separados en paquetes.

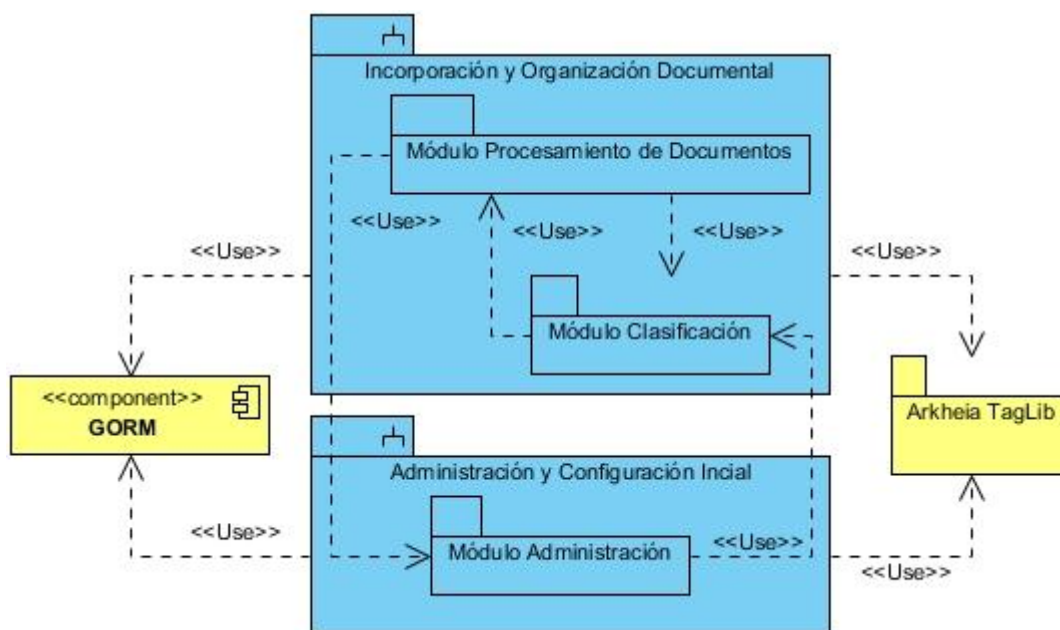


Figura 3.1. Diagrama de paquetes de componentes del sistema.

Descripción de los paquetes

En los diagramas de paquetes que se describen a continuación se utiliza el color verde para los paquetes de componentes del sistema, el color amarillo para las clases de XABAL Arkheia 3.0 que son utilizadas sin realizar modificaciones y el color azul para las clases agregadas por la solución o clases de XABAL Arkheia 3.0 a las que se realizan modificaciones.

En cada uno de los módulos están presente tanto controladores, cómo servicios, clases de dominio y vistas de la capa de presentación.

El paquete Controladores contiene las clases Groovy encargadas de gestionar las operaciones que permiten el flujo y la comunicación entre las vistas y el dominio, reciben las peticiones del usuario, ejecutan la lógica de negocio con el apoyo de los servicios y posteriormente actualizan la interfaz del sistema para que el usuario pueda ver como ha quedado el modelo de datos tras las actualizaciones.

El paquete de Servicios contiene las clases Groovy encargadas de procesar la lógica de negocio del sistema. Permitiendo aligerar la carga de los controladores en este sentido.

El paquete dominio contiene las clases Groovy que conforman el modelo de datos de la aplicación. Estas clases del dominio constituyen entidades o tablas en la base de datos y controlan la persistencia de dichas entidades durante su ciclo de vida.

El paquete Vistas contiene los componentes llamados Páginas servidor de Grails (GSP por sus siglas en inglés) encargados de generar las vistas con las cuales el usuario interactúa durante la mayor parte de su experiencia con el sistema.

Los diagramas de las figuras 3.2, 3.3 y 3.4 muestran los componentes de cada uno de estos paquetes pertenecientes a los Módulos de Administración, Clasificación y Procesamiento de Documentos respectivamente.

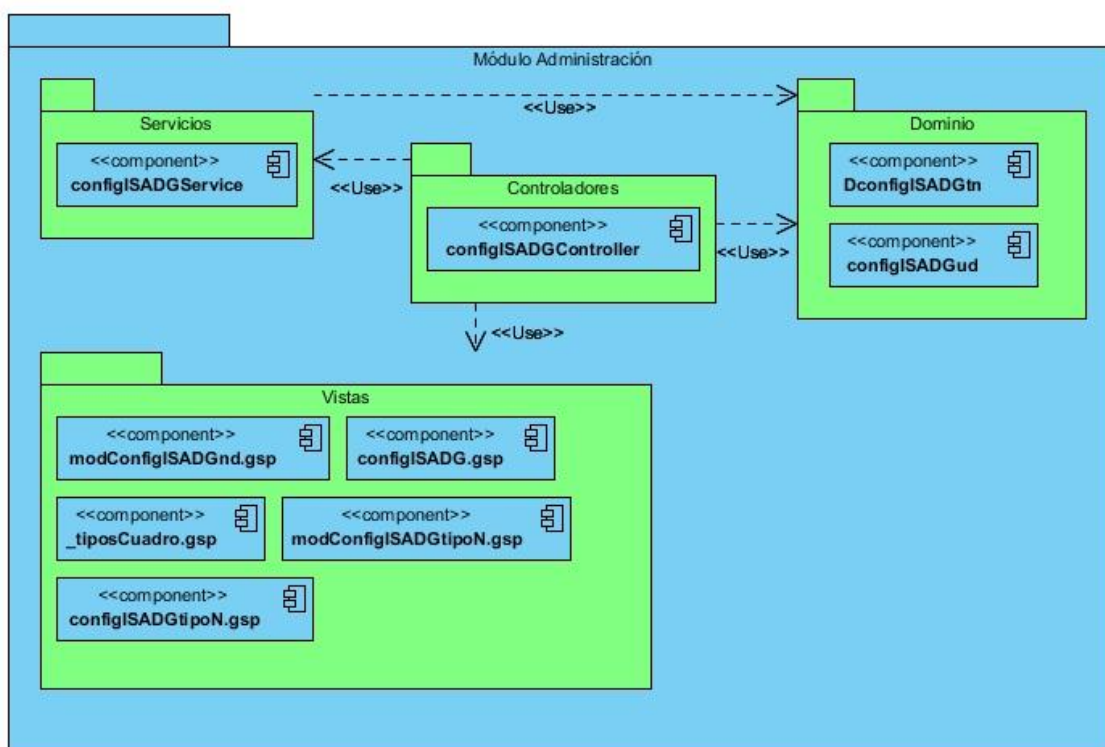


Figura 3.2. Diagrama de componentes del sistema. Módulo Administración.

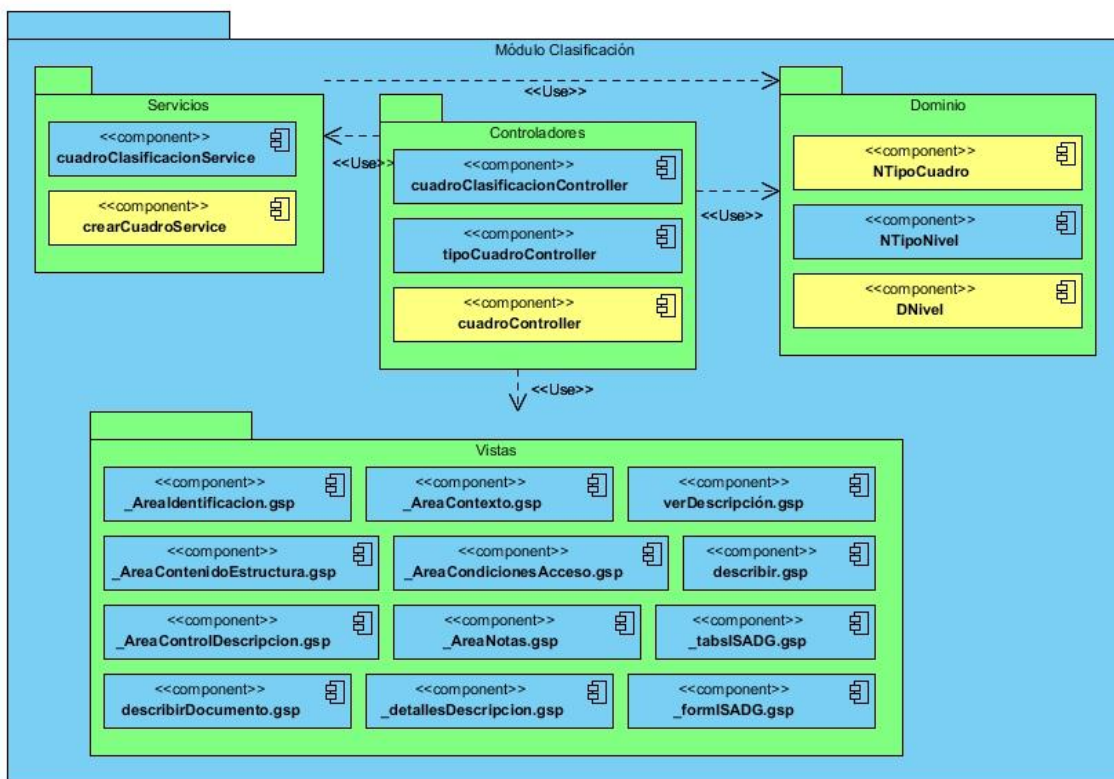


Figura 3.3. Diagrama de componentes del sistema. Módulo Clasificación.

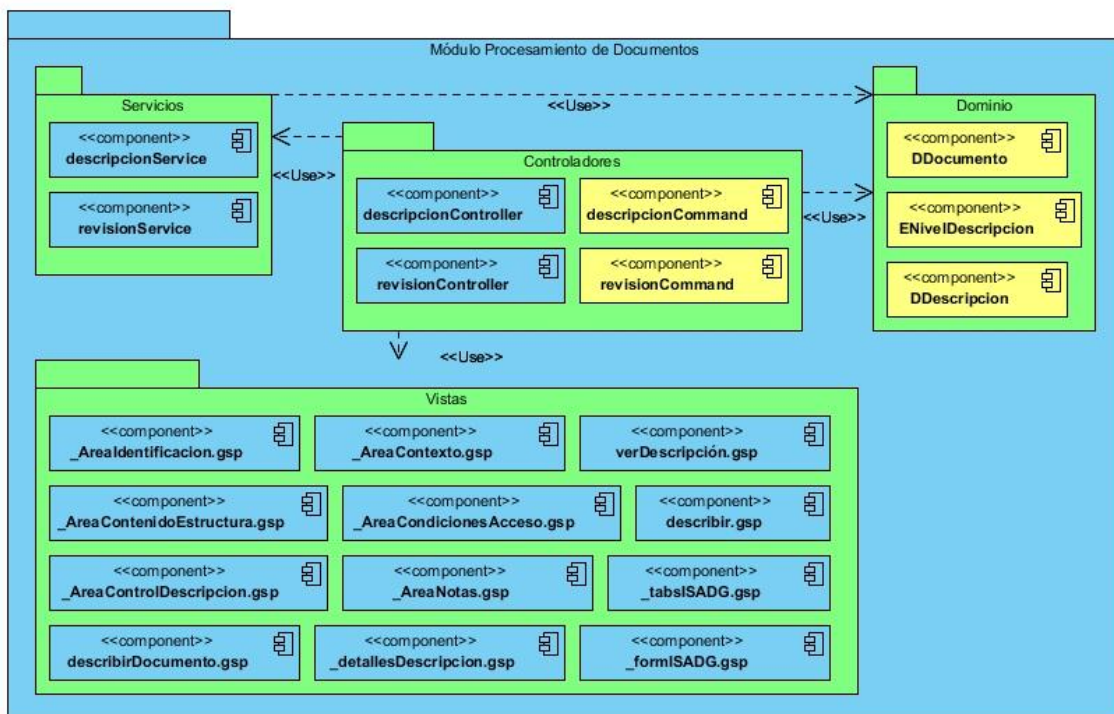


Figura 3.4. Diagrama de componentes del sistema. Módulo Procesamiento de Documentos.

3.3 Diagrama de despliegue

El diagrama de despliegue se utiliza para modelar la disposición física de los componentes de hardware utilizados en la implementación del sistema y la relación entre cada uno de ellos (Pressman 2009).

El despliegue de la propuesta de solución debe realizarse de la manera que ilustra la figura 24 para su correcto funcionamiento. Para llevarse a cabo es necesario contar con el sistema de gestión de base de datos PostgreSQL, en su versión 9.1 o una versión superior, instalado en el servidor de datos. El servidor de aplicación debe contar con el servidor web Apache Tomcat, en su versión 7.054 o una versión superior, y con la plataforma Java Runtime Environment versión 1.7.51. Las estaciones de trabajo requerirán el navegador web Mozilla Firefox en una versión igual o superior a las 10.0.

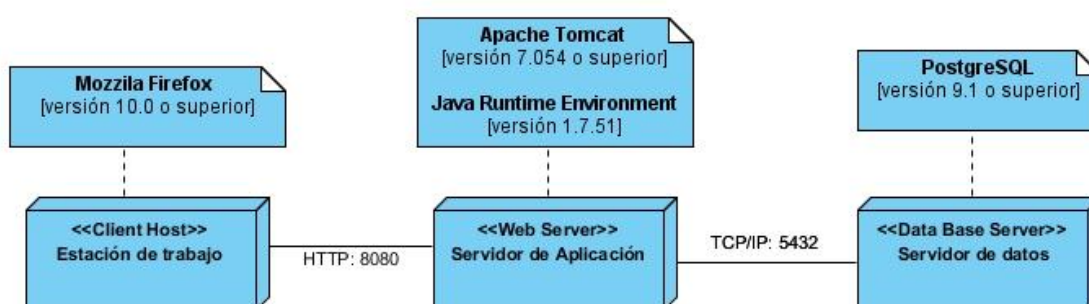


Figura 3.5. Diagrama de despliegue de la propuesta de solución.

3.4 Pruebas de *software*

Las pruebas de *software* es un concepto que a menudo, es conocido como verificación y validación del *software*. Son un elemento crítico para la garantía de calidad del *software* y representan una revisión final de las especificaciones, del diseño y de la codificación (Pressman 2009).

En este proceso se ejecutan pruebas dirigidas a componentes del *software* en específico o al sistema de *software* en su totalidad, con el objetivo de medir el grado en que se cumplen los requisitos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante técnicas de pruebas (Bolaños Alonso, Sierra Alonso y Alarcón Rodríguez 2008).

3.4.1 Estrategia de pruebas

La Estrategia de prueba es un elemento de las pruebas del *software* que establece una proyección con los pasos que se darán como parte de la prueba. Una buena práctica para estas estrategias es que se realicen de una forma tal que permitan flexibilidad para un enfoque personalizado (Pressman 2009).

La estrategia seguida para la realización de pruebas al sistema está constituida por dos niveles. En un primer nivel se realizarán pruebas de unidad, con el objetivo de verificar que cada componente se comporte como se espera sin tener en cuenta su entorno. Dichas pruebas se llevarán a cabo mediante el sistema de pruebas que pone a

disposición el framework Grails. En el segundo nivel se empleará la técnica manual denominada “Diseño de casos de prueba”, los cuales incluyen tanto pruebas de funcionalidad como de usabilidad.

3.4.2 Pruebas unitarias

Los test unitarios son aquellos en los que se verifica que un método se comporta como debería, sin tener en cuenta su entorno. Esto significa que cuando se ejecutan las pruebas unitarias de un método, Grails no inyectará ninguno de los métodos dinámicos con los que cuenta la aplicación cuando se está ejecutando, permitiendo que al trabajar con entidades o servicios el desarrollador es el responsable de crear y gestionar los objetos (Brito Calahorro 2009).

Para la correcta aplicación de las pruebas unitarias se hace necesario realizar pruebas de caja blanca al código utilizando el método del camino básico, el cual *permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez (Giraldo 2014).*

A continuación se muestra una prueba de caja blanca realizada al método configISADG de la clase controladora configISADGController:

```
def configISADG() {
    def ltn= DConfigISADGdoc.findAll()
    List<DCClasificacion> cuadros
    tipoCuadroService.eliminarNoConfigurado()
    cuadros = paginateService.init("/arkheia/web/administracion/isadg/tiposCuadro") { offset ->
        tipoCuadroService.obtenerCuadros(offset)}

    render(
        view: "/arkheia/web/administracion/isadg/configISADG",
        model: [niveles:ltn,resultList:cuadros]
    )
}
```

Figura 3.6. Método configISADG de la clase controladora configISADGController.

El análisis del método descrito en la figura 3.6 permitió elaborar el grafo de flujo que se muestra en la figura 3.7.

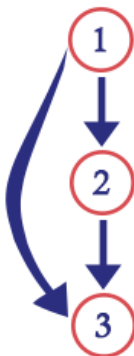


Figura 3.7. Grafo de flujo del método configISADG de la clase controladora configISADGController.

La complejidad ciclomática ($V(G)$) del grafo se calcula con la fórmula:

$$V(G) = \text{Aristas} - \text{Nodos} + 2$$

$$V(G) = 3 - 3 + 2$$

$$V(G) = 2$$

A partir de la complejidad ciclomática se obtiene la cantidad de caminos independientes con que cuenta el grafo, en este caso son los siguientes:

- 1,2,3
- 1,3

Los caminos independientes permiten diseñar casos de prueba que garanticen que todas las instrucciones del método se ejecuten al menos una vez. Los cuales fueron empleados para desarrollar las pruebas unitarias mediante el sistema de pruebas del framework Grails.

3.4.3 Casos de prueba

Un caso de prueba es un conjunto de entradas o condiciones y sus resultados esperados, bajo los cuales se determinará si el requerimiento en cuestión es parcial o completamente satisfactorio.

En los escenarios de prueba referentes a la configuración de la norma ISAD (G) resulta irrelevante el valor que se otorgue a cada variable (campos de la norma). Pues estos no alteran la respuesta del sistema. Por esta razón en las tablas 3.1 y 3.2, para una mejor visualización de las mismas, los campos de la norma fueron resumidos a una columna. En el anexo 1 se pueden encontrar los casos de prueba ampliados.

Descripción general

Permitir modificar la configuración de la norma ISAD (G) tanto para los tipos de niveles de descripción de un cuadro de clasificación como para las UDS y UDC.

Condiciones de ejecución

Para la configuración de los tipos de niveles de un cuadro de clasificación resulta necesario contar con al menos un cuadro de clasificación registrado en el sistema.

SC1 Modificar configuración ISAD (G) para unidades de descripción

Escenario	Descripción	Campos de la norma	Respuesta del sistema	Flujo central
EC 1.1 Modificar configuración UDS.	Permite modificar la configuración de la norma ISAD (G) para las UDS	V No seleccionados	Muestra un mensaje indicando que se han guardado los datos satisfactoriamente.	1. Seleccionar la opción "Norma ISAD (G)" de la opción "Administración" en el menú principal. 2. Seleccionar la opción "UDS" 3. Seleccionar la opción "Aceptar". 4. Seleccionar los campos deseados. 5. Seleccionar la opción "Salvar".

EC 1.2 Modificar configuración UDC.	Permite modificar la configuración de la norma ISAD (G) para las UDC	V Seleccionados	Muestra un mensaje indicando que se han guardado los datos satisfactoriamente.	1. Seleccionar la opción “Norma ISAD (G)” de la opción “Administración” en el menú principal. 2. Seleccionar la opción “UDC” 3. Seleccionar la opción “Aceptar”.4. Seleccionar los campos deseados. 5. Seleccionar la opción “Salvar”.
EC 1.4 Cancelar operación	Permite cancelar la configuración.	N/A	Cancela la configuración realizada.	1. Seleccionar la opción “Norma ISAD (G)” de la opción “Administración” en el menú principal. 2. Seleccionar la opción “UDC” 3. Seleccionar la opción “Aceptar”. 4. Seleccionar la opción “Cancelar”.

Tabla 3.1. Escenario de prueba Modificar configuración ISAD (G) para unidades de descripción.

SC2 Modificar configuración ISAD (G) para los tipos de nivel de un cuadro de clasificación.

Escenario	Descripción	Campos de la norma	Respuesta del sistema	Flujo central
EC 1.1 Modificar configuración para tipo de nivel.	Permite modificar la configuración de la norma ISAD (G) para un tipo de nivel de un cuadro de clasificación.	V Seleccionados	Muestra un mensaje indicando que se han guardado los datos satisfactoriamente.	1. Seleccionar la opción “Norma ISAD (G)” de la opción “Administración” en el menú principal. 2. Seleccionar la opción “Configurar ISAD (G) para el cuadro de clasificación deseado” 3. Seleccionar el tipo de nivel deseado 4. Seleccionar la opción “Aceptar”. 5. Seleccionar los campos deseados. 6. Seleccionar la opción “Salvar”.
EC 1.2 Cancelar operación	Permite cancelar la configuración.	N/A	Cancela la configuración realizada.	1. Seleccionar la opción “Norma ISAD (G)” de la opción “Administración” en el menú principal. 2. Seleccionar la opción “Configurar ISAD(G) para el cuadro de clasificación deseado” 3. Seleccionar el tipo de nivel deseado 4. Seleccionar la opción “Aceptar”. 5. Seleccionar la opción “Cancelar”.

Tabla 3.2. Escenario de prueba Modificar configuración ISAD (G) para unidades de descripción.

3.4.4 Resultados de las pruebas aplicadas

El reporte que muestra la figura 3.8 fue generado por Grails luego de ser aplicadas las pruebas unitarias. Certificando la correcta ejecución de dichas pruebas.

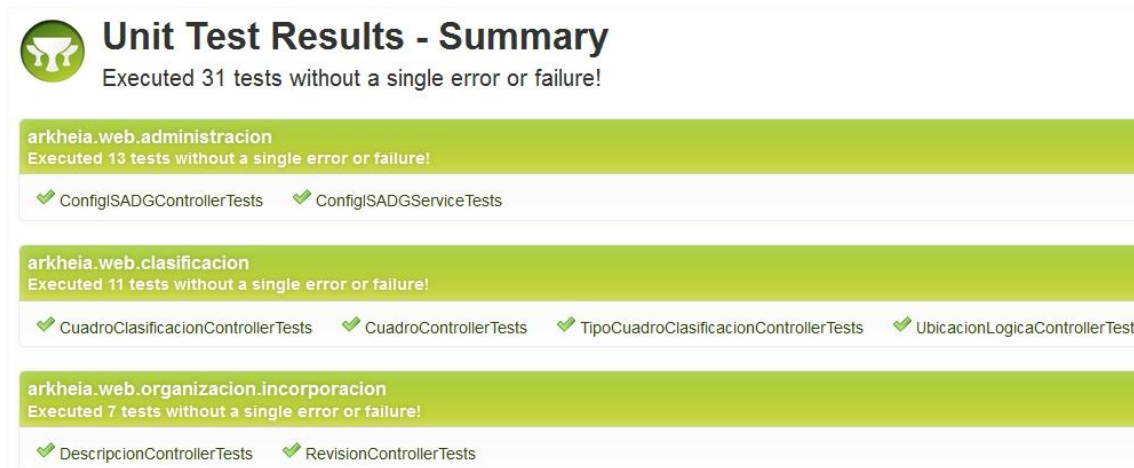


Figura 3.8. Resumen de las pruebas unitarias y de integración realizadas.

La figura 3.9 muestra los resultados obtenidos durante las pruebas funcionales aplicadas al sistema en tres iteraciones. En la primera iteración fueron identificadas doce no conformidades de las cuales siete fueron resueltas para la segunda iteración. En la tercera y última iteración de pruebas no se detectaron no conformidades.

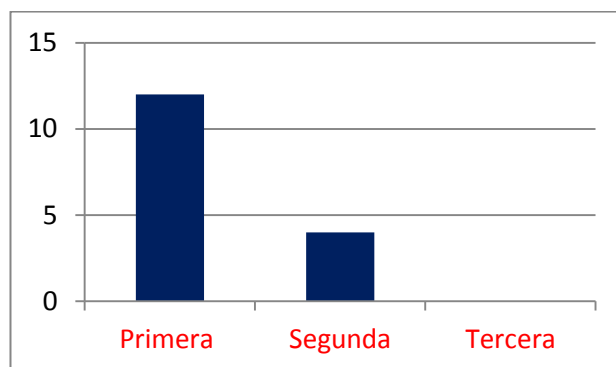


Figura 3.9. Resultados obtenidos mediante la aplicación de pruebas funcionales.

3.5 Conclusiones del capítulo

La elaboración del diagrama de componentes de *software* permitió obtener una visión general de la distribución de la propuesta mediante la identificación de los componentes necesarios para la implementación de la solución así como la forma en que estos se relacionan entre sí. El diagrama de despliegue a su vez identifica los elementos necesarios para garantizar el funcionamiento del sistema en un entorno de trabajo real, así como los protocolos de comunicación utilizados entre dichos elementos y los requerimientos que los mismos deben cumplir para un funcionamiento óptimo.

El diseño y la puesta en práctica de la estrategia de prueba permitieron comprobar el correcto cumplimiento de los requerimientos del sistema.

Conclusiones Generales

Con la culminación del presente trabajo de diploma se arriba a las siguientes conclusiones:

- Utilizando como base el marco teórico referencial y a partir del estudio realizado sobre los sistemas de gestión de archivos históricos presentes en el mercado nacional e internacional se decidió llevar a cabo la incorporación de la configuración de la norma ISAD (G) para garantizar una correcta descripción multinivel en el sistema XABAL Arkheia 3.0.
- La utilización de las métricas TOC y RC permitió validar el diseño elaborado para garantizar el cumplimiento de los atributos de calidad.
- Con el desarrollo de la configuración de la norma ISAD (G) para el sistema XABAL Arkheia se logra una correcta descripción multinivel de los fondos documentales mejorando considerablemente la organización de los mismos y la experiencia de los usuarios con el sistema.
- Con la aplicación de las pruebas unitarias y la técnica manual de casos de prueba se corroboró el correcto funcionamiento de la solución propuesta.

Recomendaciones

- Se recomienda implementar en el sistema XABAL Arkheia la posibilidad de exportar e importar la estructura de los cuadros de clasificación incluyendo la configuración de sus niveles.
- Se recomienda a los archiveros responsables, realizar la configuración de la norma ISAD (G) para las unidades de descripción simples y compuestas justo después de completar el proceso de instalación de XABAL Arkheia 3.0.
- En el caso de los tipos de niveles de los cuadros de clasificación, realizar la configuración de la norma después de crear cada cuadro de clasificación, evitando así describir innecesariamente y la pérdida de información.

Bibliografía Referenciada

- Archeevo. *KEEP SOLUTIONS* [en línea], 2016. [Consulta: 25 enero 2017]. Disponible en: <http://www.keep.pt/en/produtos/archeevo/>.
- ARCINIEGAS HERRERA, J.L., 2001. *Architectural patterns regarding web application domain usability*. Universidad Nacional de Colombia: s.n.
- BOLAÑOS ALONSO, D., SIERRA ALONSO, A. y ALARCÓN RODRÍGUEZ, I., 2008. *Pruebas de software y JUnit*. Madrid: Educación. ISBN 978-84-8322-354-3.
- BRITO CALAHORRO, N., 2009. *Manual de desarrollo web con Grails*. S.I.: Imaginaworks. ISBN 978-84-613-2651.
- BUNGE, M., 2000. *La investigación científica: su estrategia y su filosofía*. S.I.: Siglo XXI. ISBN 978-968-23-2225-9.
- CERIA, S., 2005. *Casos de Uso. Un Método Práctico para Explorar Requerimientos*. S.I.:
- CIGED, 2014. *Ayuda Sistema de gestión de archivos históricos XABAL ARKHEIA 3.0*. 2014. S.I.: s.n.
- CONSEJO INTERNACIONAL DE ARCHIVOS, 2000. *ISAD(G): Norma Internacional General de Descripción Archivística*. 2000. S.I.: s.n.
- CRUZ MUNDET, J.R., 1996. *Manual de archivística*. 2da. Madrid: Ediciones Pirámide. Biblioteca del Libro, 930.25. ISBN 83-368-0860-6.
- DRAKE, J.M., 2008. *Análisis de requisitos y especificación de una aplicación*. Santander.
- GACETA OFICIAL, 2009. *Decreto-Ley No. 265 del Sistema Nacional de Archivos de la República de Cuba*. 5 mayo 2009. S.I.: s.n. 482.
- GIRALDO, M., 2014. *Camino Básico. Prezi* [en línea]. [Consulta: 19 mayo 2017]. Disponible en: <https://prezi.com/jvbhsculhy2k/camino-basico/?webgl=0>.
- GONZÁLEZ GARCÍA, P., 1997. *Archivo General de Indias*. p 1997. S.I.: Madrid: Dirección General del Libro, Archivos y Bibliotecas: Lunwerg.
- KEITH ROCHER, G., 2006. *Definite Guide to Grails*. S.I.: Apress. ISBN 978-1-59059-758-3.
- LARMAN, C., 2003. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Segunda. S.I.: Pearson Educación. 9788420534381
- LONDOÑO PALACIO, O.L., MALDONADO GRANADOS, L.F. y CALDERÓN VILLAFÁÑEZ, L.C., 2014. *Guías para construir estados del arte*. [en línea], [Consulta: 8 junio 2017]. Disponible en: <http://repositorio.minedu.gob.pe/handle/123456789/4637>.

Manual de validación. Validación de requisitos. , 2004. S.l.: Universidad Politécnica de Madrid.

Norma ISO UNE 15489, 2006. 2006. S.l.: s.n.

PITTI, D.V., 1999. Encoded Archival Description: An Introduction and Overview. *D-Lib Magazine*, vol. 5, no. 11.

PRESSMAN, R.S., 2009. *Ingeniería del Software. Un enfoque práctico*. Madrid: McGraw-Hill. ISBN 970-10-5473-3.

RAUSCHMAYER, A., 2014. *Speaking JavaScript* [en línea]. S.l.: O'Reilly Media, Inc. [Consulta: 19 mayo 2017]. ISBN 978-1-4493-6503-5.

RODRÍGUEZ SÁNCHEZ, T., 2015. *Metodología de desarrollo para la Actividad productiva de la UCI*. 6 marzo 2015. S.l.: UCI.

SÁNCHEZ, J., 2003. JavaScript. Manual de referencia. [en línea]. S.l.: [Consulta: 19 mayo 2017].

SMITH, G. y LEDBROOK, P., 2014. *Grails in Action*. S.l.: s.n. ISBN 978-1-61729-096-1.

SOMMERVILLE, I., 2011. *Ingeniería de Software*. 9. Madrid: Pearson Educación, S.A. 84-7829-074-5

SPIRO, L., 2009. Archival Management Software. A Report for the Council on Library and Information Resources. . Washington, D.C.: Council on Library and Information Resources.

STEINMARK, C., 1996. The use of information technology in the european search - rooms : results from the danish pilot project Ouverture. *Journal of the Society of Archivists*, vol. 17, no. 1.

UDD, C.M. y F. N, J., 2008. *Beginning Groovy and Grails: from novice to professional*. California: California. ISBN 978-1-4302-1045-0.

Bibliografía Consultada

- ACEVEDO GUTIÉRREZ, F.O., 2004. Introducción a la organización de archivos. . S.I.
- ALVARADO, A.V.T., 2011. EL SISTEMA DE ARCHIVO Y GESTIÓN DE DOCUMENTOS DE LA UNIVERSIDAD CENTRAL DE VENEZUELA. UNA PROPUESTA. *PontodeAcesso*, vol. 4, no. 3, pp. 95–112.
- Archeevo. *KEEP SOLUTIONS* [en línea], 2016. [Consulta: 25 enero 2017]. Disponible en: <http://www.keep.pt/en/produos/archeevo/>.
- ARCINIEGAS HERRERA, J.L., 2001. *Architectural patterns regarding web application domain usability*. Universidad Nacional de Colombia: s.n.
- ARRIBAS DEL POZO, M., 2014. *Gestión de Archivos*. S.I.: Paraninfo. ISBN 978-84-283-9828-2.
- BOLAÑOS ALONSO, D., SIERRA ALONSO, A. y ALARCÓN RODRÍGUEZ, I., 2008. *Pruebas de software y JUnit*. Madrid: Educación. ISBN 978-84-8322-354-3.
- BRITO CALAHORRO, N., 2009. *Manual de desarrollo web con Grails*. S.I.: Imaginaworks. ISBN 978-84-613-2651.
- BUNGE, M., 2000. *La investigación científica: su estrategia y su filosofía*. S.I.: Siglo XXI. ISBN 978-968-23-2225-9.
- CERIA, S., 2005. Casos de Uso. Un Método Práctico para Explorar Requerimientos. . S.I.:
- CIGED, 2014. *Ayuda Sistema de gestión de archivos históricos XABAL ARKHEIA 3.0*. 2014. S.I.: s.n.
- CONSEJO INTERNACIONAL DE ARCHIVOS, 2000. *ISAD(G): Norma Internacional General de Descripción Archivística*. 2000. S.I.: s.n.
- CRUZ MUNDET, 2012. *ARCHIVISTICA: GESTION DE DOCUMENTOS Y ADMINISTRACION DE ARCHIVOS*. S.I.: ALIANZA EDITORIAL. ISBN 978-84-206-0952-2.
- CRUZ MUNDET, J.R., 1996. *Manual de archivística*. 2da. Madrid: Ediciones Pirámide. Biblioteca del Libro, 930.25. ISBN 83-368-0860-6.
- CRUZ MUNDET, J.R., 2011. *Diccionario de Archivística*. Madrid. España: Alianza Editorial. ISBN 978-84-206-5285-6.
- CRUZ MUNDET, J.R., 2014. *Archivística: El libro universitario*. S.I.: Alianza Editorial. ISBN 978-84-206-8925-8.
- DORADO SANTANA, Y. y MENA MUGICA, M.M., 2009. Evolución de la ciencia archivística. *Acimed*, vol. 20, no. 1, pp. 0–0.

- DRAKE, J.M., 2008. Análisis de requisitos y especificación de una aplicación. . Santander.
- FELIU, M.C.M., 1995. ISAD (G): hacia un estándar internacional de descripción archivística. *MÉI: Métodos de Información*, vol. 2, no. 8, pp. 26–31.
- FOGUS, M., 2013. *Functional JavaScript*. S.I.: O'Reilly Media. ISBN 978-1-4493-6072-6.
- GACETA OFICIAL, 2009. *Decreto-Ley No. 265 del Sistema Nacional de Archivos de la República de Cuba*. 5 mayo 2009. S.I.: s.n. 482.
- GAUCHI RISSO, V., 2012. Aproximación teórica a la relación entre los términos gestión documental, gestión de información y gestión del conocimiento. *Revista española de Documentación Científica*, vol. 35, no. 4, pp. 531-554. ISSN 1988-4621, 0210-0614. DOI 10.3989/redc.2012.4.869.
- GONZÁLEZ GARCÍA, P., 1997. *Archivo General de Indias*. p 1997. S.I.: Madrid: Dirección General del Libro, Archivos y Bibliotecas: Lunwerg.
- KEITH ROCHER, G., 2006. *Definite Guide to Grails*. S.I.: Apress. ISBN 978-1-59059-758-3.
- KÖNIG, D., KING, P., LAFORGE, G., D'ARCY, H., CHAMPEAU, C., PRAGT, E. y SKEET, J., 2015. *Groovy in Action*. Segunda Edición. S.I.: s.n. ISBN 978-1-935182-44-3.
- KRUCHTEN, P., 2000. *The Rational Unified Process An Introduction*. 2. S.I.: Addison Wesley. ISBN 0-201-70710-1.
- LARMAN, C., 2003. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Segunda. S.I.: Pearson Educación. 9788420534381
- LEDO, V., JOSEFINA, M., PÉREZ, A. y BÁRBARA, A., 2012. Gestión de la información y el conocimiento. *Educación Médica Superior*, vol. 26, no. 3, pp. 474-484. ISSN 0864-2141.
- LINDLEY, C., 2012. *JavaScript Enlightenment*. S.I.: O'Reilly Media. ISBN 978-1-4493-4287-6.
- LÓPEZ CORTINA, G. y COBAS PARADA, J.A., 2011. *Módulo para el Intercambio de Descripciones entre Sistemas de Gestión de Archivos Históricos Archivenhis*. La Habana, Cuba: Universidad de las Ciencias Informáticas.
- Manual de validación. Validación de requisitos. , 2004. S.I.: Universidad Politécnica de Madrid.
- MOTA, M. de la A.S. y RUIPÉREZ, M.G., 1999. *El patrimonio documental: fuentes documentales y archivos*. S.I.: Univ de Castilla La Mancha. ISBN 978-84-8427-023-2.
- PITTI, D.V., 1999. Encoded Archival Description: An Introduction and Overview. *D-Lib Magazine*, vol. 5, no. 11.

- PRESSMAN, R.S., 2009. *Ingeniería del Software. Un enfoque práctico*. Madrid: McGraw-Hill. ISBN 970-10-5473-3.
- PRIMORAC, C.R., MARIÑO, S.I. y GODOY, M.V., 2012. TIC y Sociedad: Especificación de Requisitos para Apoyar la Gestión de Información. *Revista Digital Sociedad de la Información. Monográfico TIC* [en línea], no. 33. [Consulta: 26 octubre 2016]. Disponible en: <http://sociedadelainformacion.com/34/ticsociedad.pdf>.
- RAUSCHMAYER, A., 2014. *Speaking JavaScript* [en línea]. S.I.: O'Reilly Media, Inc. [Consulta: 19 mayo 2017]. ISBN 978-1-4493-6503-5.
- RODRÍGUEZ SÁNCHEZ, T., 2015. *Metodología de desarrollo para la Actividad productiva de la UCI*. 6 marzo 2015. S.I.: UCI.
- SÁNCHEZ, J., 2003. JavaScript. Manual de referencia. [en línea]. S.I.: [Consulta: 19 mayo 2017].
- SÁNCHEZ, J.C., 2012a. *La investigación científica y tecnológica*. S.I.: Ediciones Díaz de Santos. ISBN 978-84-9969-389-7.
- SÁNCHEZ, J.C., 2012b. *Los métodos de investigación*. S.I.: Ediciones Díaz de Santos. ISBN 978-84-9969-391-0.
- SCOTT, S., 2008. *Groovy Recipes*. S.I.: The Pragmatic Programmers. ISBN 978-0-9787392-9-4.
- SMITH, G. y LEDBROOK, P., 2014. *Grails in Action*. S.I.: s.n. ISBN 978-1-61729-096-1.
- SOMMERVILLE, I., 2005. *Ingeniería de Software*. 7ma. Madrid: Pearson Educación, S.A. 84-7829-074-5
- SORIANO SINFONTES, R. y FLEITES AVILA, Y., 2015. *Módulo Procesamiento de Audiovisuales para el sistema XABAL Arkheia 2.1 para la OAHCE*. La Habana, Cuba: Universidad de las Ciencias Informáticas.
- SOZA LÓPEZ, D., 2013. *Modelos de Casos de Uso del Sistema*.
- Standard ECMA-262*, 2016. 7 th Edition. S.I.: ECMA International.
- STEINMARK, C., 1996. The use of information technology in the european search - rooms : results from the danish pilot project Ouverture. *Journal of the Society of Archivists*, vol. 17, no. 1.
- SUBRAMANIAM, V., 2008. *Programming Groovy*. S.I.: The Pragmatic Programmers. ISBN 978-1-934356-09-8.
- UDD, C.M. y F. N, J., 2008. *Beginning Groovy and Grails: from novice to professional*. California: California. ISBN 978-1-4302-1045-0.
- VISBAL, A. y M, S., 2009. La gestión documental, de información y el conocimiento en la empresa: El caso de Cuba. *ACIMED*, vol. 19, no. 5, pp. 0-0. ISSN 1024-9435.
- VIVAS MORENO, A., 2013. *Breve Historia Cultural de la Archivística*. S.I.: Abecedario. ISBN 978-84-9978-160-0.

ZAKAS, N.C., 2012. *Maintainable JavaScript*. First Edition. S.I.: O'Reilly Media, Inc. ISBN 978-1-4493-2768-2.

ZAZO, J.L.B., 2000. La normalización: base del análisis documental en los archivos. *Scire: representación y organización del conocimiento*, vol. 6, no. 1, pp. 55–75.