



Universidad de las Ciencias
Informáticas

**CENTRO DE INFORMATIZACIÓN UNIVERSITARIA
FACULTAD 1**

Módulo Recuperación de Información para el
Sistema de Gestión de Documentos Históricos Dexcriba.

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

Autores: María Isabel González Del Monte
Carlos Lázaro Jiménez Puerto

Tutores: Ing. Leodán De los Ángeles Buduén.
Ing. Marlon Jorge Remedios González

Ciudad de La Habana

Junio 2013

Pensamiento



**No puedes preguntarle a los consumidores qué quieren
y luego pretender dárselo.
En el tiempo que has estado fabricándolo,
ellos querrán una cosa nueva.**

Steve Jobs

1955 - 2011



Declaración de autoría

Por este medio declaro que soy el único autor de este trabajo y autorizo al Centro de Informatización Universitaria de la Universidad de las Ciencias Informáticas para que hagan el uso que estimen pertinente con este trabajo.

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

Firma del tutor

Ing. Leodán de los Ángeles Buduén

Firma del tutor

Ing. Marlon J. Remedios González

Firma del autor

María Isabel González del Monte

Firma del autor

Carlos Lázaro Jiménez Puerto

Ing. Leodán De los Ángeles Buduén: Graduado de Ingeniero en Ciencias Informáticas en el año 2009, con siete años de experiencia en temas de desarrollo de software¹ de gestión de documentos históricos y participación en diferentes eventos relacionados con la materia. En la actualidad labora en el Departamento de Gestión Documental de CENIA de la Universidad de las Ciencias Informáticas (UCI).

Ing. Marlon Jorge Remedios González: Graduado como Ingeniero en Ciencias Informáticas en el año 2008 en la Universidad de las Ciencias Informáticas (UCI). Actualmente alcanza la categoría docente de Profesor Asistente. Forma parte de la matrícula de la maestría en informática aplicada de la UCI 3era edición. En el 2011 acreditó una publicación y participación en evento en Informática 2011. Cuenta con participación además en REV 2012 que será publicado en IAOE.

¹ Conjunto de programas de cómputo, documentos asociados y esquemas de configuración necesarios para que estos programas operen.



Agradecimientos

Quiero agradecer ante todo a mi familia por permitirme crecer entre tanto cariño, por creer en mí, incluso cuando nadie más lo hace. Por protegerme solo lo necesario y dejar que cometa mis propios errores. Sin ustedes, este sueño no hubiese sido posible.

A mi mamá, por permitirme estudiar donde siempre quise, por haber sido madre y padre para mí, ten por seguro que si hoy estoy aquí es por ti.

A mi tío José, por haber sido más que un tío, por ser la figura a imitar, por darme el ejemplo y demostrarme que por muy largo que sea el camino siempre se llega a la meta.

A mis abuelos **Celia y Gil** por haber puesto todo su empeño en mi crianza, por darme el cariño de los 4 abuelos.

A mis hermanos **Sheyla y José Carlos**, por ese cariño incomparable que me llena de vida y me da fuerzas para seguir adelante, hoy le doy gracias a la vida por tenerlos a mi lado.

A mi prima **Nayeli**, por ser más que mi prima: mi hermana también, gracias por ser tan especial, y por la ternura que desprenden tus palabras cuando me dices "tata".

A mi madrina **Olguita**, gracias por permitirme ser tu hijo y acompañarme en cada momento que hasta hoy he vivido.

A Yusmany por cuidar de mi familia en todo este largo período de tiempo, por querer a mi familia como si fuese la suya.

A mi tía **Mary** por confiar en mí e implusarme a seguir adelante, este título también constituye tu victoria.

A mi mamá **Migdalia**, gracias por cuidar de mí y enseñarme tantas cosas buenas.

A Yuniel por su apoyo incondicional, por haber aprendido en tan corto tiempo a comprenderme y sacar lo mejor de mí.

A los amigos de aquí y de allá:

Lianet, Andy, Yanet, Marbelis, Daimara, Yaima, Luis Angel, Luisbel, Dayán, Exxon y Yerandy.

Gracias por haberme ofrecido su amistad sincera.

A Yaili, por mantenerse cerca siempre, por ser mi consuelo y mi fuerza en momentos que los necesitaba, por saber verdaderamente el concepto de la palabra amigo.

A Maide, Yenli, Anelé, Raúl y Yosmel gracias por estar ahí, por tener un trato diferente conmigo, por apoyarme y decirme: todo estará bien, aún cuando yo sabía que todo estaba perdido.

Gracias también a **Maikel**, que llegó último, pero se que se el muchachón promete.

A todos los profesores que durante estos cinco años me han hecho un mejor profesional.

A los que ya no son parte de mi vida, pero que influyeron mucho en ella.

A todos los que de una forma u otra han contribuido a la realización de este sueño.

A todos muchas gracias.

Carlos Lázaro Jiménez Puerto



A la **Caridad del Cobre** por ser mi guía y protectora todos estos años.

A mi Papá **Rene** por confiar en mi y apoyarme en todas mi desiciones.

A mi **mamá** por darme la fuerza para seguir cuando yo creí a que no podía más.

A mi mejor amigo en todo el mundo **Angel** por siempre estar a mi lado en los buenos y malos momentos,
gracias por ser un amigo incondicional.

A mis hemanos **Rene** y **Lisi** por quererme.

A mi tía **Alicia** por ser mi segunda mamá.

A mis abuelos **Nene, Ada**, papi **Abel** y mami **María** por darme a los mejores padres del mundo.

A mi novio **Manuel**, por estar a mi lado y darme apoyo en los momentos difíciles,
además por esperarme estos años y confiar en mi. Te amo bb.

A **Maylen** (MX) por ser mi mejor amiga y acompañarme en todos mi momentos de alegrías y tristezas.

A mi piquete **Tay, Victor, Hilario, Jota** gracias a todos por darme muchos momentos de alegrías.

A **Carlos**, creo que no me pude tener un mejor compañero de tesis, me encantó trabajar contigo.

A **Leodan** nuestro tutor por estar tan atento a nuestro trabajo y preocuparse.

A las chicas de mi apartamento **Leidy, Islen, Evelyn, Diana y Macdemis**,
gracias por darme la oportunidad de poder vivir con todas ustedes.

A **Jordan, Migue y Mary** por acogerme y hacerme parte de su familia gracias.

A toda mi familia.

A todos lo que de una forma u otra me ayudaron en cualquier momentos.
Gracias.

María Isabel González del Monte



Dedicatoria

A mis padres:

Quiero dedicarle esta tesis a mis padres que con su luz iluminan todos mis caminos, por siempre estar ahí cuando más sola me he sentido, por hacer de mí, por encima de cualquier circunstancia una persona de bien, de buenos principios y costumbres. Gracias por siempre creer en mí, por toda esta vida de cariño y de enseñanzas.

María Isabel González del Monte

A mi mamá Celita.

Por haberme apoyado en todo momento, por saber el significado de la palabra "madre", por la motivación constante que me ha permitido ser una persona de bien.

A mis familiares.

A mis abuelos maternos, a mi tío Jose del cual aprendí aciertos en momentos difíciles; a mi tía Maria Antonia, a Sheyla, José Carlos y Nayeli por ser las personas que me motivan a darles lo mejor de mí y me hacen marcar el camino a seguir. A ustedes les dedico el logro de hacerme Ingeniero y todos mis éxitos en esta vida, la enseñanza moral e intelectual.

A un gran amigo, Osvaldo:

Me dolió dentro el vacío de tu ausencia, aún sin haberte vuelto a ver después de una conversación inconclusa, y en mi recuerdo afloró cada momento en que supiste decir lo que verdaderamente necesitaba(a tu manera claro). Una lágrima porque te fuiste, y mil lágrimas de alegría mientras te encontrabas entre nosotros. A ti también te dedico esta tesis.

Carlos Lázaro Jiménez Puerto



Resumen

Como consecuencia de la creciente presencia de las Tecnologías de la Información y las Comunicaciones (TIC) en la vida diaria, y la necesidad de consultar documentos importantes, que por su antigüedad o nivel de deterioro no podían ser consultados físicamente surgen los sistemas de gestión de documentos históricos. Dexcriba es uno de estos sistemas que organiza, controla y gestiona digitalmente el ciclo de vida de los documentos de un archivo, pero no cuenta con un mecanismo que permita la recuperación del fondo documental contenido en el software, por lo que los usuarios no pueden obtener directamente información para el estudio o trabajo propio de cada área del sistema, así como la tramitación de los mismos.

El desarrollo de este trabajo posibilitó la construcción de un módulo que permite la búsqueda y recuperación de las descripciones contenidas en el sistema, sobre seis campos de búsqueda definidos como obligatorios. Los subsistemas de búsqueda sencilla, avanzada y exploratoria permiten al usuario una localización personalizada a partir del uso del ordenamiento y filtrado de los resultados, la navegación sobre la estructura jerárquica del sistema y la posibilidad de exportar los resultados que luego pueden ser cargados para continuar con la investigación.

Fue desarrollado utilizando el framework CodeIgniter, PostgreSQL como Sistema Gestor de Bases de Datos y guiado por RUP con CMMI nivel II.

Palabras claves: búsqueda, descripción de documentos, recuperación de información.



Abstract

Due to the growing presence of Information Technologies and Communications (ICT) on daily life, and the need of consulting important documents, that for its age or level of impairment could not be consulted physically arise Historical document management systems. Dexcriba is one such system that organizes, controls and and manages in a digital way the document life cycle, but does not have a recovery mechanism for the documentary, basement contained in the software, so that users can not get directly information for study or work specific to each area of the system, ss well as their processing.

The development of this work enabled the construction of a module that allows search and retrieval of the descriptions contained in the system, about six search fields defined as required. Simple search subsystems, Advanced and exploratory allow to the user exploratory a personalized location based on the use of sorting and filtering of the results, navigation on the tree that simulates the hierarchical structure of the system and the ability to export the results which can then be loaded to continue the investigation.

It was developed using the framework CodeIgniter, PostgreSQL Manager System Databases and guided CMMI Level II of RUP.

Keywords: search, description of documents, information retrieval.



Índice de contenidos

| | |
|---|-----------|
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA..... | 7 |
| 1.1. RECUPERACIÓN DE LA INFORMACIÓN..... | 7 |
| 1.1.1. <i>Recuperación de información y recuperación de datos</i> | 10 |
| 1.2. GESTIÓN ARCHIVÍSTICA. CONCEPTOS FUNDAMENTALES Y CARACTERÍSTICAS..... | 11 |
| 1.3. RELEVANCIA Y PERTINENCIA DE LOS RESULTADOS | 16 |
| 1.4. PROCESOS DE BÚSQUDA EN SISTEMAS DE GESTIÓN DE DOCUMENTOS HISTÓRICOS..... | 17 |
| 1.4.1. <i>ICA-AtoM</i> | 17 |
| 1.4.2. <i>Albalá</i> | 17 |
| 1.4.3. <i>Archon</i> | 18 |
| 1.4.4. <i>Archivo 3000</i> | 18 |
| 1.4.5. <i>Portal Pares</i> | 18 |
| 1.4.6. <i>ArchiVenHis</i> | 19 |
| 1.5. ANÁLISIS DE LAS SOLUCIONES ENCONTRADAS | 19 |
| 1.6. TECNOLOGÍAS NECESARIAS PARA LA IMPLEMENTACIÓN DEL MÓDULO..... | 20 |
| 1.6.1. <i>Lenguajes de programación</i> | 20 |
| 1.6.1.1. Lenguaje del lado del servidor: PHP v5.3..... | 20 |
| 1.6.1.2. Lenguajes del lado del cliente: <i>JavaScript v 1.8</i> | 21 |
| 1.6.1.3. Lenguajes de Marcado | 22 |
| 1.6.1.3.1. Lenguaje de Marcado de Hipertexto (HTML) v4.01..... | 23 |
| 1.6.1.3.2. Lenguaje de Etiquetado Extensible (XML) | 23 |
| 1.6.1.4. CSS v2.1..... | 24 |
| 1.6.2. <i>Servidor web: Apache v2.2</i> | 24 |
| 1.6.3. <i>Lenguaje de modelado:</i> | 25 |
| 1.6.3.1. Lenguaje Unificado de Modelado (UML) | 25 |
| 1.6.4. <i>Framework de desarrollo.</i> | 26 |
| 1.6.4.1. CodeIgniter v2.1.3..... | 26 |
| 1.6.4.2. jQuery v1.8.3..... | 26 |
| 1.6.5. <i>Herramientas de programación</i> | 27 |
| 1.6.5.1. Entorno de desarrollo integrado (IDE): Netbeans v7.2 | 28 |
| 1.6.6. <i>Sistema gestor de base de datos</i> | 28 |
| 1.6.6.1. PostgreSQL v9.1 | 29 |
| 1.6.7. <i>Herramientas CASE</i> | 29 |
| 1.6.7.1. Visual Paradigm v8.0..... | 30 |
| 1.7. METODOLOGÍA DE DESARROLLO DE SOFTWARE | 30 |
| 1.8. CONCLUSIONES PARCIALES | 31 |
| CAPÍTULO 2. CARACTERÍSTICAS DEL MÓDULO RECUPERACIÓN DE LA INFORMACIÓN..... | 33 |
| 2.1. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN | 33 |
| 2.2. MODELO DE DOMINIO | 33 |
| 2.2.1. <i>Descripción de los principales conceptos</i> | 33 |
| 2.2.2. <i>Diagrama de clases del dominio</i> | 34 |
| 2.3. ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE..... | 34 |
| 2.3.1. <i>Técnicas para la captura de requisitos</i> | 35 |
| 2.3.1.1. Análisis de sistemas existentes..... | 35 |
| 2.3.1.2. Juego de roles..... | 35 |
| 2.3.1.3. Entrevista..... | 35 |
| 2.3.2. <i>Requerimientos funcionales</i> | 36 |
| 2.3.3. <i>Requisitos no funcionales</i> | 37 |



| | |
|---|-----------|
| 2.4. MODELO DE CASOS DE USO..... | 38 |
| 2.4.1. Actores | 38 |
| 2.4.2. Diagrama de casos de uso del sistema..... | 39 |
| 2.4.3. Validación de requerimientos: Matriz de trazabilidad | 39 |
| 2.4.4. Descripción de los casos de uso..... | 40 |
| 2.5. ANÁLISIS DEL SISTEMA..... | 40 |
| 2.5.1. Diagramas de clases del análisis. | 41 |
| 2.5.2. Diagramas de interacción | 41 |
| 2.6. DISEÑO DEL SISTEMA | 42 |
| 2.6.1. Diagramas de clases del diseño | 42 |
| 2.7. DISEÑO DE LA BASE DE DATOS..... | 43 |
| 2.7.1. Modelo de datos | 43 |
| 2.8. ARQUITECTURA | 44 |
| 2.8.1. Modelo-Vista-Controlador (MVC) | 44 |
| 2.8.2. Patrones de diseño..... | 45 |
| 2.8.2.1. Patrones GOF..... | 47 |
| 2.9. MANEJO DE ERRORES..... | 48 |
| 2.10. ESTÁNDAR DE CODIFICACIÓN | 48 |
| 2.11. CONCLUSIONES PARCIALES..... | 49 |
| CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DEL MÓDULO RECUPERACIÓN DE LA INFORMACIÓN | 50 |
| 3.1. DIAGRAMA DE COMPONENTES..... | 50 |
| 3.2. MODELO DE DESPLIEGUE | 51 |
| 3.3. PRUEBAS..... | 52 |
| 3.3.1. Pruebas de Caja Blanca..... | 52 |
| 3.3.1.1. Diseño de casos de prueba de caja blanca | 53 |
| 3.3.2. Pruebas de Caja Negra..... | 54 |
| 3.3.2.1. Diseño de casos de prueba de caja negra..... | 55 |
| 3.3.3. PRUEBAS DE CARGA Y STRESS..... | 56 |
| 3.3.3.1. Resultados pruebas de carga y stress..... | 56 |
| 3.3.4. Resultado de las pruebas realizadas. | 57 |
| 3.4. CONCLUSIONES PARCIALES..... | 59 |
| CONCLUSIONES GENERALES | 60 |
| RECOMENDACIONES | 61 |
| REFERENCIAS BIBLIOGRÁFICAS | 62 |
| BIBLIOGRAFÍA..... | 67 |
| ANEXOS | 68 |



Índice de tablas

| | |
|---|----|
| Tabla 1: Descripción del actor del sistema | 39 |
| Tabla 2: Matriz de trazabilidad | 39 |
| Tabla 3: Descripción del CUS Realizar búsqueda avanzada | 40 |
| Tabla 4: Caminos identificados para las pruebas de caja blanca | 54 |
| Tabla 5: Caso de prueba para el camino básico #1 | 54 |
| Tabla 6: : Caso de prueba para el camino básico #2 | 54 |
| Tabla 7: : Caso de prueba para el camino básico #3 | 54 |
| Tabla 8: Diseño de casos de prueba para el CUS Realizar búsqueda sencilla | 56 |
| Tabla 9: Resultados pruebas de carga y stress | 57 |
| Tabla 10: Resultados de las pruebas realizadas..... | 58 |
| Tabla 11: Comparación entre los resultados obtenidos por iteración | 58 |
| Tabla 12: Formato de la entrevista aplicada..... | 68 |
| Tabla 13: Descripción del CUS Realizar búsqueda avanzada | 69 |
| Tabla 14: Descripción del CUS Realizar búsqueda exploratoria | 69 |
| Tabla 15: Descripción del CUS Ordenar resultados de las búsquedas | 70 |
| Tabla 16: Descripción del CUS Exportar resultados de las búsquedas | 70 |
| Tabla 17: Descripción del CUS Importar resultados de las búsquedas | 71 |
| Tabla 18: Descripción de la entidad descripción correspondiente al modelo Descripción | 75 |
| Tabla 19: Descripción de la entidad descripcion_productor correspondiente al modelo Descripción | 75 |
| Tabla 20: Descripción de la entidad productor correspondiente al modelo Descripción | 75 |
| Tabla 21: Descripción de la entidad tipo_entidad correspondiente al modelo Descripción | 76 |
| Tabla 22: Descripción de la entidad tipo_nivel correspondiente al modelo Descripción..... | 76 |
| Tabla 23: Descripción de la entidad posible_hijo_nivel correspondiente al modelo Descripción | 76 |
| Tabla 24: Enumeración del método búsqueda sencilla | 77 |
| Tabla 25: Valores de entrada de datos para el caso de prueba del CUS Realizar búsqueda avanzada | 77 |
| Tabla 26: Diseño de casos de prueba para el CUS Realizar búsqueda avanzada..... | 79 |
| Tabla 27: Diseño de casos de prueba para el CUS Exportar resultados de la búsqueda | 79 |
| Tabla 28: Diseño de casos de prueba para el CUS Importar resultados de la búsqueda | 79 |
| Tabla 29: Diseño de casos de prueba para el CUS Realizar búsqueda exploratoria | 79 |
| Tabla 30: Diseño de casos de prueba para el CUS Ordenar resultados de la búsqueda..... | 80 |



Índice de figuras

Ilustración 1: Diagrama de clases del dominio del módulo Recuperación de la Información34

Ilustración 2: Diagrama de CUS39

Ilustración 3: Diagrama de clases del análisis del CUS Realizar búsqueda sencilla41

Ilustración 4: Diagrama de interacción del CUS Realizar búsqueda sencilla42

Ilustración 5: Diagrama de clases del diseño del módulo Recuperación de la información43

Ilustración 6: Diagrama de componentes global del módulo Recuperación de la información50

Ilustración 7: Paquete de Vistas51

Ilustración 8: Paquete de Controladoras51

Ilustración 9: Paquete de Modelos51

Ilustración 10: Modelo de despligue52

Ilustración 11: Grafo de flujo53

Ilustración 12: Diagrama de clases del análisis del CUS Realizar búsqueda avanzada72

Ilustración 13: Diagrama de clases del análisis del CUS Realizar búsqueda exploratoria72

Ilustración 14: Diagrama de clases del análisis del CUS Exportar resultados de la búsqueda72

Ilustración 15: Diagrama de clases del análisis del CUS Importar resultados de la búsqueda72

Ilustración 16: Diagrama de interacción del CUS Realizar búsqueda avanzada73

Ilustración 17: Diagrama de interacción del CUS Realizar búsqueda exploratoria73

Ilustración 18: Diagrama de interacción del CUS Exportar resultados de la búsqueda73

Ilustración 19: Diagrama de interacción del CUS Importar resultados de la búsqueda74

Ilustración 20: Modelo de datos del esquema Descripción74



Introducción

Desde tiempos remotos el hombre se ha preocupado por mantener guardadas sus vivencias y recuerdos. De esta manera, surgieron diferentes medios para grabar estos conocimientos, tales como: el arte rupestre, los jeroglíficos, la escritura cuneiforme, y el uso de la tinta y el papiro. Posteriormente con el descubrimiento del papel la humanidad obtuvo una nueva forma para expresar y hacer perdurar sus pensamientos; siendo estas las bases para la confección de los documentos, libros y artículos que se conocen en la actualidad. Con el paso del tiempo y la necesidad cada vez más creciente de que estos conocimientos pudiesen ser heredados por sucesivas generaciones fue designado el Archivo o fondo documental como un lugar con la finalidad de que las personas almacenaran y coleccionaran sus creaciones.

Un archivo es una institución o parte estructural de ella que reúne, conserva, ordena y difunde conjuntos orgánicos de documentos para la gestión administrativa, la información, la investigación y la cultura (Fuster, 2006).

Los archivos remontan sus orígenes al surgimiento de la escritura. Diferentes culturas contaron con importantes archivos que sirvieron como una herramienta para el control de su población y sus riquezas, entre ellas la griega, la romana y la egipcia. Hoy los archivos tienen como finalidad la preservación de documentos que poseen valor administrativo, legal, fiscal, científico, económico, político, cultural y/o histórico, documentos que obviamente ostentan una importancia vital a la hora de intentar investigar en la identidad y la reconstrucción histórica de una nación, de modo que se garantice la integridad y paso a futuras generaciones. La naturaleza de los archivos, los principios de su organización y conservación, así como los medios para su utilización son aspectos estudiados por la ciencia denominada Archivística (Mena, 2009).

Analizando las consideraciones que anteceden, se puede visualizar que la Archivística es un complejo campo del conocimiento, que requiere para su elaboración científica el manejo ordenado y coherente de los preceptos que se encuentren en vigor en un lugar y época determinados.

Este manejo, así como el logro de altos grados de eficiencia en el mismo, necesitan la utilización de instrumentos técnicos para el almacenamiento, procesamiento y recuperación de datos e información, para lo cual la aplicación de la informática ha sido de gran ayuda.



El fenómeno de la informática ha venido desarrollándose de forma vertiginosa, convirtiéndose en una herramienta indispensable en cualquier área de estudio o trabajo. Bajo esta perspectiva, la sociedad ha venido impulsando la emergencia y difusión de nuevas tecnologías adaptables prácticamente a todas las necesidades de información de cada usuario, por lo que difícilmente es posible prever límites a la expansión de este importante instrumento técnico.

Como consecuencia de la creciente presencia de las Tecnologías de la Información y las Comunicaciones (TIC) en la vida diaria, se hizo necesaria la búsqueda de alternativas para que algunos documentos importantes que por su antigüedad o nivel de deterioro no podían ser consultados físicamente, pudiesen ser estudiados y permitieran a los usuarios interesados adquirir información referente, por esta razón surgen los sistemas de gestión documental (GD), con el fin de ser utilizados en instituciones que posean grandes volúmenes de información objeto de conservación, pues no solo se garantiza el buen estado de los documentos sino que se agiliza el proceso de búsqueda de los mismos.

Día a día el ser humano se adapta con un ritmo creciente a la utilización de estos sistemas para llevar un registro de sus actividades cotidianas. Pero si se presta atención a este punto, puede notarse que estas actividades crecen no solo en complejidad y frecuencia, sino en la cantidad de documentos que se genera. Tomando en cuenta lo anterior se deduce que las bases de datos de los sistemas crecen al mismo ritmo que se va dando esta rápida evolución en la sociedad.

Por este motivo los sistemas de gestión documental han prevalecido a la luz del crecimiento del volumen de reseñas, referencias, detalles, registros y notas referentes a los documentos, que han provocado la obsolescencia de los métodos tradicionales de almacenamiento y búsqueda. Las descripciones de los documentos han dejado de ser generales para entrar en detalle y su cantidad, por lo tanto, ha ido en aumento, provocando una labor más pronunciada en los últimos veinte años.

Los sistemas de gestión de documentos históricos (SGDH) son una particularidad de los anteriormente mencionados y se orientan a la configuración de un cuerpo de información contenido en una o más bases de datos relacionados con cualquiera de las áreas temáticas de la Archivística, mediante un diálogo interactivo entre el usuario y la máquina. La eficiencia y la utilidad constituyen características concretas que hacen relevante esta herramienta para el archivero², que propicia la satisfacción de necesidades de

² Un archivero o archivista es una persona con titulación superior o media dedicada a la organización y mantenimiento de un archivo público o privado.



información del usuario, a través del uso de mecanismos y técnicas de recopilación, organización, conservación y difusión de datos históricos.

En los últimos años, varios países del mundo han desarrollado proyectos en materia de informática aplicada a la gestión de documentos históricos, a través de los cuales se pretende agilizar los procesos de obtención, procesamiento de datos y una regulación informática más efectiva.

Actualmente en la Universidad de las Ciencias Informáticas (UCI) se desarrolla un sistema de gestión de documentos históricos llamado Dexcriba, que cuenta con módulos de digitalización, descripción, conservación, ubicación y control de acceso, pero no cuenta con uno que permita la recuperación de información.

Este sistema contiene descripciones, imágenes y procesos relacionados con el ciclo de vida de los documentos simulando la estructura de los catálogos tradicionales. Ante estos grandes catálogos y organizaciones jerárquicas bastante amplias, la búsqueda es una herramienta fundamental, ya que muchos usuarios tienen claro lo que necesitan, y prefieren buscarlo directamente.

La tendencia actual, frente a la tradicional búsqueda y recuperación en que hacía falta la labor de bibliotecarios y documentalistas que realizaban la indización de forma manual, es que el usuario sea capaz, ofreciéndole las herramientas adecuadas, de hacer las búsquedas por sí solo.

Teniendo en cuenta el creciente uso de aplicaciones web, se ha propiciado el uso de las búsquedas como un mecanismo de ayuda al usuario para localizar lo que verdaderamente necesita, por lo que introducir palabras claves y presionar el botón buscar resulta innato.

Por lo anteriormente planteado se propone como **problema a resolver**: ¿Cómo garantizar la recuperación de información en un archivo histórico?

Se define como **objeto de estudio** de esta investigación la recuperación de información en sistemas de gestión de documentos.

De acuerdo con la problemática planteada se propone como **objetivo general**: desarrollar un módulo para el Sistema de Gestión de Documentos Históricos Dexcriba que permita la recuperación de información sobre los documentos custodiados en las instituciones siguiendo las normas establecidas por el Consejo Internacional de Archivos.



El **campo de acción** de esta investigación estará enmarcado hacia la recuperación de información en sistemas de gestión de documentos históricos.

Con el propósito de cumplir con el objetivo trazado, se plantean las siguientes **tareas de la investigación**:

- ✓ Estudio de las normas y estándares existentes para la descripción de documentos de archivo. Permitirá la identificación de los campos más significativos que pudiesen ser utilizados como referencia para las búsquedas.
- ✓ Evaluación de las características y el funcionamiento de los principales sistemas de gestión de documentos históricos existentes que realizan las búsquedas sobre las descripciones de los documentos de archivos. Permitirá desechar las deficiencias encontradas y tomar de ellos lo mejor para integrarlos en un solo sistema.
- ✓ Especificación de los requisitos funcionales y no funcionales para el módulo a desarrollar. Esto permitirá la descripción y el comportamiento esperado en el *software* una vez desarrollado, partiendo de la identificación de las necesidades del negocio, así como la interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requisitos del *software*.
- ✓ Elaboración del modelo del sistema, lo cual permitirá a los involucrados manejar un vocabulario común para lograr un mejor entendimiento del contexto en que se sitúa el sistema.
- ✓ Diseño de las interfaces de usuario. Brindará una aproximación al diseño visual que contendrá las funcionalidades identificadas con anterioridad.
- ✓ Diseño e implementación del modelo de datos necesario para todos los subsistemas del módulo. Posibilitará identificar si es necesario o no la implementación de un modelo de datos para el módulo, en este caso se definirán qué tablas serán utilizadas para recuperar la información necesaria.
- ✓ Implementación de pruebas a cada uno de los subsistemas desarrollados, que brindará una valoración real y por separado de cada uno de estos subsistemas, luego en el módulo y finalmente de forma integral para comprobar que el *software* realice correctamente las tareas indicadas en la especificación del problema.



- ✓ Ejecución y análisis de los resultados arrojados por las pruebas de caja blanca y caja negra. Permitirá hacer un estudio de los principales resultados obtenidos durante las pruebas y ver en qué medida afectan la solución del problema general.
- ✓ Integración de los subsistemas. Permitirá verificar si el sistema además de cumplir los requisitos que se plantean se integra correctamente a la arquitectura del sistema general.

Para la realización de este trabajo de tesis se utilizaron diferentes métodos científicos, ellos constituyen un “Conjunto de reglas que señalan el procedimiento para llevar a cabo una investigación” (Riveros, 2000). Este conjunto de reglas parten de principios claros, razonables e incuestionables, que servirán para dar validez a las reglas del método científico.

Los métodos teóricos a utilizar son:

- ✓ **Análisis histórico-lógico** que permitirá estudiar el modo en que han evolucionado los estándares y normas para la descripción de documentos.
- ✓ **Analítico-sintético** el cual posibilitará hacer un estudio de los principales Sistemas de Gestión de Documentos Históricos, así como las tendencias en la búsqueda y recuperación de la información, para posteriormente sintetizar y aplicar el conocimiento adquirido, de modo que se logre un mayor entendimiento y se pueda arribar a conclusiones que contribuyan a la solución del problema planteado.
- ✓ **Modelación:** Se utilizará para lograr una mejor comprensión de los procesos asociados a la recuperación de la información, a partir de la representación en el modelo de dominio de entidades y actores.

Entre los métodos empíricos que se emplearán están:

- ✓ **Observación**, que guiará el estudio del estado del arte, permitiendo realizar un análisis sistémico, selectivo y objetivo de los principales sistemas que en la actualidad pueden realizar recuperación de información contenida en ellos.
- ✓ **Entrevista** no estructurada con la intención de obtener información referente a los procesos de búsqueda, así como criterios de expertos en el tema.



Con la realización del presente trabajo de diploma se pretende lograr una mayor difusión del patrimonio documental resguardado por las instituciones, a partir del uso de Dexcriba como sistema genérico capaz de adaptarse a diferentes entornos y situaciones de trabajo que cuente con diferentes mecanismos de búsquedas que permitan recuperar y filtrar la información para que el usuario pueda acceder de forma directa a la que le interesa.

El presente trabajo de diploma está estructurado de la siguiente manera:

Capítulo 1. Fundamentación teórica: se expondrá un conjunto de conceptos y criterios fundamentales para lograr un mayor entendimiento de la recuperación de información, la gestión documental y la archivística. Se estudiarán los principales sistemas de gestión de documentos históricos para analizar tendencias en las búsquedas que realizan y la exploración de archivos a partir de estas. Además, se expondrán las distintas tecnologías a utilizar en el desarrollo del presente trabajo de diploma, así como la metodología de desarrollo de *software* a utilizar.

Capítulo 2. Características del módulo Recuperación de la información: en este capítulo se realizará una descripción detallada de la solución que se propone para cumplir el objetivo de la presente investigación. Se representarán los principales conceptos que se manejan en el contexto del sistema mediante un modelo de dominio. Será realizada una especificación de requisitos funcionales y no funcionales, basado en el levantamiento de las necesidades que debe satisfacer la solución. A partir de los requisitos obtenidos, se definirán los casos de uso y los actores que se relacionan con cada uno de ellos, confeccionando el diagrama de casos de uso del sistema, con la correspondiente descripción de cada uno de ellos. Se llevará a cabo el análisis y diseño del sistema, la representación del modelo de datos, los patrones de arquitectura y diseño que serán utilizados.

Capítulo 3. Implementación y pruebas del módulo Recuperación de la información: se representarán los elementos físicos necesarios para un correcto despliegue de la aplicación, empleando para ello un diagrama de despliegue. Se efectuará la validación de la solución de acuerdo con los requisitos que debe cumplir para garantizar una calidad óptima, a través de pruebas funcionales.



Capítulo 1. Fundamentación teórica

En este capítulo se abordan algunos de los conceptos que constituyen la base de la investigación, aspectos teóricos que posibilitan una mejor comprensión del problema y el proceso de búsqueda de la solución. Se estudian conceptos relacionados con la gestión documental y la archivística en general. Se presentan los resultados del estudio del estado del arte, exponiendo antecedentes y actualidad del tema que se trata en la investigación. Se exponen los resultados de un estudio realizado sobre los procesos de búsqueda y obtención de la información en sistemas de gestión de documentos históricos. Quedan descritos además algunos elementos como la metodología de desarrollo del *software* y tecnologías a utilizar en el desarrollo de la aplicación.

1.1. Recuperación de la información.

El concepto de recuperación de información (RI), a pesar de ser un término bastante empleado presenta cierta confusión a la hora de establecer una definición que lo sitúe adecuadamente dentro del campo de las ciencias de la información.

Rijsbergen es el autor que mejor introduce este problema al considerar que se trata de un término que suele ser definido en un sentido muy amplio (Rijsbergen, 1999). En realidad, el profuso uso de este término, al igual que ocurre en otras disciplinas con otros vocablos que también pueden parecer básicos, ha propiciado que el mismo no se encuentre bien empleado en muchas ocasiones, ya que unas veces los autores lo presentan como sinónimo de la recuperación de datos (RD) llevada desde la perspectiva de las bases de datos. Otro conjunto de autores expresan las diferencias que, en su opinión, presentan ambos conceptos (con lo cual la definición de recuperación de información queda, en cierto modo, supeditada a la anterior), un tercer grupo de autores la define de forma muy genérica sin entrar en mayores consideraciones sobre estas diferencias, y un cuarto y último grupo pasa de largo sobre este problema, profundizando más en la explicación de los sistemas de recuperación de información (SRI).

El primer grupo de definiciones están muy influenciadas por la tecnología informática, cuya evolución ha inducido a considerar sinónimos ambos conceptos, llegándose a olvidar que se puede recuperar información sin procedimientos informáticos (aunque no es lo más común hoy en día). Aun así, el frecuente y necesario empleo de una tecnología no debe sustituir el adecuado uso de los conceptos terminológicos. Un claro ejemplo de este desacierto es el Glosario de la Asociación de Bibliotecarios Americanos, que define el término "*information retrieval*" como recuperación de la información en primera



acepción y como recuperación de datos en una segunda acepción (Glosario A.L.A., 1983), considerando ambos términos sinónimos en Lengua Inglesa³. Igualmente, el Diccionario Mac Millan de Tecnología de la Información considera la recuperación de información como las “técnicas empleadas para almacenar y buscar grandes cantidades de datos y ponerlos a disposición de los usuarios” (Longley, 1989)

Un segundo grupo de autores fijan diferencias. Meadow piensa que la recuperación de la información es “una disciplina que involucra la localización de una determinada información dentro de un almacén de información o base de datos” (Meadow, 1993). Este autor, implícitamente, establece que la recuperación de información se encuentra asociada con el concepto de selectividad, ya que la información específica ha de extraerse siguiendo algún tipo de criterio discriminatorio (selectivo por tanto). Pérez-Carballo y Strzalkowski redundan en esta tesis: “una típica tarea de la recuperación de información es traer documentos relevantes desde un gran archivo en respuesta a una pregunta formulada” (Pérez, 2000). Del mismo modo, Grossman y Frieder indican que recuperar información es “encontrar documentos relevantes, no encontrar simples correspondencias a unos patrones de bits” (Grossman, 1998). Meadow considera que no es lo mismo la recuperación de información entendida como traducción del término inglés *information recovery* que cuando se traduce el término *information retrieval*, ya que “en el primer caso no es necesario proceso de selección alguno” (Meadow, 1993).

De similar opinión es Blair, quien dedica gran parte de la presentación de su libro *Language and representation in information retrieval* a asentar las diferencias entre *data retrieval* e *information retrieval*, utilizando como criterios distintivos, entre otros (Blair, 1990):

- ✓ En recuperación de datos se emplean preguntas altamente formalizadas, cuya respuesta es directamente la información deseada. En cambio, en recuperación de información las preguntas resultan difíciles de trasladar a un lenguaje normalizado y la respuesta será un conjunto de documentos que probablemente contendrá lo deseado, con un evidente factor de indeterminación.
- ✓ Según la relación entre el requerimiento del sistema y la satisfacción de usuario, la recuperación de datos es determinista y la recuperación de información es posibilista, debido al nivel de incertidumbre presente en la respuesta.

³ Este Glosario indica que “document retrieval” es un término sinónimo de “information retrieval”.



- ✓ En recuperación de datos el criterio a emplear es la exactitud de lo encontrado, mientras que en recuperación de información, el criterio de valor es el grado en el que la respuesta satisface las necesidades de información del usuario, es decir, su percepción personal de utilidad.

Baeza-Yates plantea las diferencias entre ambos tipos de recuperación con argumentos quizá algo menos abstractos que los anteriormente empleados por otros autores, destacando que “los datos se pueden estructurar en tablas, árboles, etc. para recuperar exactamente lo que se quiere, el texto no posee una estructura clara y no resulta fácil crearla” (Baeza-Yates, 1999).

Baeza-Yates se preocupa especialmente de las estructuras de datos y métodos de acceso a los mismos, siendo este autor una verdadera referencia en esta materia. Curiosamente, a la hora de definir la recuperación de información, en lugar de proponer una definición propia, emplea la elaborada por Salton: “la recuperación de la información tiene que ver con la representación, almacenamiento, organización y acceso a los *ítem* de información” (Salton, 1983).

El tercer grupo de autores emplea la definición formulada por Salton (base de la mayoría de definiciones de a bibliografía especializada), añadiendo el rasgo diferenciador en que estos autores no profundizan en escrutar las diferencias entre recuperación de datos y recuperación de información, bien por no ser objeto de sus trabajos o por considerarlas suficientemente establecidas en trabajos previos.

Croft estima que la recuperación de información es el conjunto de tareas mediante las cuales el usuario localiza y accede a los recursos de información que son pertinentes para la resolución del problema planteado. En estas tareas desempeñan un papel fundamental los lenguajes documentales, las técnicas de resumen y la descripción del objeto documental (Croft, 1987). Tramullas Saz impregna su definición del carácter selectivo comentado anteriormente al afirmar que “el planteamiento de la recuperación de información en su moderno concepto y discusión, hay que buscarlo en la realización de las pruebas de Cranfield y en la bibliografía generada desde ese momento y referida a los mecanismos más adecuados para extraer, de un conjunto de documentos, aquellos que fuesen pertinentes a una necesidad informativa dada” (Tramullas, 1997).

El cuarto y último grupo de autores se distinguen porque eluden definir la recuperación de la información. Su máximo exponente a Chowdhury, en su libro *Introduction to modern information retrieval* señala que “el término recuperación de la información fue acuñado en 1952 y fue ganando popularidad en la comunidad científica de 1961 en adelante”, mostrando después los propósitos, funciones y componentes de los SRI



(Chowdhury, 1999). Otro autor perteneciente a esta corriente es Korfhage, quien se centra en el almacenamiento y recuperación de la información, considerando a estos procesos como las dos caras de una moneda. Para este autor, “un usuario de un sistema de información lo utiliza de dos formas posibles: para almacenar información en anticipación de una futura necesidad, y para encontrar información en respuesta a una necesidad” (Korfhage, 1997).

De lo anteriormente expuesto se resume que la recuperación de la información es el arte y/o ciencia que se encarga de la búsqueda y presentación de información relevante, de grandes colecciones de documentos, a un usuario que hace una petición normalmente en lenguaje natural.

1.1.1. Recuperación de información y recuperación de datos

Del latín *datum* (lo que se da), un dato es definido como un documento, una información o un testimonio que permite llegar al conocimiento de algo o deducir las consecuencias legítimas de un hecho.

La información se describe como un mensaje normalmente bajo la forma de un documento o algún tipo de comunicación audible o visible (Santos, 2009).

A diferencia de los datos, la información tiene significado (relevancia y propósito). Los datos se convierten en información cuando su creador les añade significado.

El conocimiento es una mezcla de experiencia, valores, información y “saber hacer” que sirve como marco para la incorporación de nuevas experiencias e información, y es útil para la acción (Santos, 2009).

La recuperación de datos se basa en determinar qué documentos en una colección contienen las palabras clave expresadas en una consulta. Esto muchas veces no satisface las necesidades de información del usuario.

La forma en la que se interpretan las consultas en la recuperación de datos es muy estricta, de forma que la aparición de un documento que no incluya los términos expresados en la consulta se considera como un fracaso, mientras que esto no es así en la recuperación de información. Esto se debe a que los resultados en la recuperación de información pretenden ser relevantes y en la recuperación de datos deben ser acertados.



De hecho, las consultas de los usuarios en un sistema de RI pueden ser interpretadas con cierto grado de ambigüedad, lo que genera respuestas con una correspondencia parcial a la pregunta, mientras que en RD las consultas se interpretarán de forma literal y producirán resultados exactos.

La diferencia fundamental entre ambas alternativas reside en el hecho de que la Recuperación de Información trabaja sobre la base del lenguaje natural, mientras que la Recuperación de Datos trabaja con datos con una estructura y semántica muy bien establecidas.

1.2. Gestión archivística. Conceptos fundamentales y características

El término Archivo procede del latín *archivum*, aunque su origen más remoto se encuentra en la lengua griega y puede traducirse como “residencia de los magistrados”. El término se utiliza para nombrar al conjunto ordenado de documentos que una sociedad, una institución o una persona elabora en el marco de sus actividades y funciones (Barroso, 2010).

Esta palabra se utiliza generalmente para designar a aquel lugar que tiene como finalidad la recopilación y conservación de documentos, usualmente producidos en otro lugar y como resultado de la concreción de sus respectivas actividades, que pueden ser documentos, libros, recortes de diarios viejos, entre otros y que ostentan una importancia vital a la hora de intentar explorar en la identidad y la reconstrucción histórica de una institución o nación.

Según la visión de Mayra Mena Mugica “el archivo no es más que el reflejo natural y la plasmación en sus documentos de las actividades y tareas de una entidad determinada” (Mena, 2005).

Antonia Heredia define un archivo como “uno o más conjuntos de documentos, sea cual sea su fecha, su forma y soporte material, acumulados en un proceso natural por una persona o institución pública o privada en el transcurso de su gestión, conservados, respetando aquel orden, para servir como testimonio e información para la persona o institución que los produce, para los ciudadanos o para servir de fuentes de historia” (Heredia, 1991).

En el 2011 Cruz Mundet ofrece una definición un poco más actualizada, definiendo el archivo como “un sistema corporativo de gestión que contribuye de manera efectiva, mediante una metodología propia, a la definición de los procesos de producción administrativa garantizando la correcta creación de los documentos, su tratamiento, conservación, acceso y comunicación” (Cruz, 2011).



De manera general un archivo puede definirse como un conjunto de documentos, sin importar su naturaleza, soporte o fecha de creación que son acumulados por una persona o institución en el desarrollo de su actividad, y sirven como información o testimonio de sus creadores. Así mismo se entienden también por archivos las instituciones culturales donde se reúnen, conservan, ordenan y difunden conjuntos de documentos utilizables en la investigación, la cultura, la información y la gestión administrativa.

Otros de los conceptos que guarda una estrecha relación con los archivos, es el de documentos de archivo, que constituyen sin lugar a dudas un elemento fundamental para la existencia de los archivos y son definidos como “el testimonio material de un hecho o acto realizado en el ejercicio de sus funciones por personas físicas o jurídicas, públicas o privadas, de acuerdo con las características de tipo material y formal” según lo expresado por Mayra Mena Mugica (Mena, 2005).

El documento de archivo se define en el Diccionario de Terminología Archivística como el documento que contiene una información de cualquier fecha, forma y soporte material, producido o recibido por cualquier persona física o moral y por una institución pública o privada en el ejercicio de su actividad (Cruz, 2011).

De esta manera, se concluye definir que son documentos producidos o recibidos por una persona o institución en el cumplimiento de sus funciones y que se generan en el curso de su gestión o actividad para el cumplimiento de sus objetivos y son conservados para que puedan ser utilizados como prueba o fuente de información.

Los archivos y documentos de archivo constituyen el objeto de estudio de la archivística, ciencia relativamente nueva que tuvo su surgimiento a mediados del siglo XIX. Desde entonces muchos teóricos han tratado de ofrecer una definición exacta de esta ciencia.

Para Mundet la definición correcta es la dada por el Consejo Internacional de Archivos (C.I.A.) que se encuentra en el diccionario de terminología Archivística: "el estudio teórico y práctico de los principios, procedimientos y problemas concernientes a las funciones de los archivos". Aquí engloba dentro de los archivos a los documentos y los pone al mismo nivel de importancia (Cruz, 2011).

En cambio, la definición que da Heredia es contraria a la de Cruz Mundet, ya que para Heredia la archivística es la ciencia de los archivos, no de los documentos, aunque en última instancia estos sean el producto integrante de aquellos. Por lo tanto considera que la función primordial de la Archivística serán



los archivos, poniendo en un segundo plano a los documentos. Así, su definición definitiva de Archivística es: "la ciencia que estudia la naturaleza de los archivos, los principios de su conservación y organización y los medios para su utilización" (Heredia, 1991).

Según el Diccionario de Terminología, archivística es "la disciplina que trata de los aspectos teóricos y prácticos (tipología, organización, funcionamiento, planificación, etcétera) de los archivos y el tratamiento archivístico de sus fondos documentales" (Ministerio de Cultura, 1995).

En resumen, la archivística puede ser definida como la ciencia que estudia la naturaleza de los archivos, los medios para su utilización, así como los principios que rigen su organización y conservación.

Una vez comprendidos estos conceptos es necesario puntualizar que existe un enlace creado por el archivero entre la información contenida en los documentos y las personas que desean consultarla. Para ello el mismo debe realizar un análisis de todos los documentos y sintetizar la información que poseen, generando descripciones (representaciones de las unidades documentales) para ofrecerlas a los interesados (Gómez, 2005).

"La descripción archivística es el medio utilizado por el archivero para obtener información contenida en los documentos y ofrecerla a los interesados en ella" (Heredia, 1991). Se realiza con el objetivo de brindar información a los usuarios y facilitar el trabajo al archivero. La descripción que se posee debe ser concreta, breve y contener las ideas básicas; tanto los factores internos como los externos deben ser descritos.

La **Norma Internacional General de Descripción Archivística ISAD (G)** es uno de los instrumentos descriptivos utilizados por los archiveros como guía para este proceso.

Esta norma constituye una guía general para la elaboración de descripciones archivísticas. Está compuesta por un conjunto de reglas generales encaminadas a asegurar la creación de descripciones coherentes, pertinentes y explícitas, facilitar la recuperación y el intercambio de información sobre los documentos de archivo, compartir los datos de autoridad y hacer posible la integración de descriptores de diferentes archivos en un sistema unificado de información; no contempla la forma de presentar los elementos de descripción, por lo que los instrumentos de descripción suelen ser muy variados. Su finalidad en general es dar información sobre la documentación de un archivo (total o parcial) para localizar y recuperar dicha documentación y se caracteriza por ser una norma, con una macroestructura



muy definida pero muy flexible en cuanto a nivel de descripción, profundidad de descripción y detalle de descripción de los diferentes elementos.

El número de niveles de descripción depende del volumen, complejidad del fondo, demandas de información de los usuarios, valor histórico de los documentos, etc. Las cinco unidades de descripción (documento o conjunto de documentos objeto de descripción) básicas según la norma ISAD (G) (32) son: Fondo, Subfondo, Serie, Expediente y la Unidad documental.

En cuanto a la profundidad en la descripción, la ISAD (G) establece 26 elementos descriptivos, de los cuales señala 6 como esenciales: código de referencia, título, productor, fechas, volumen de la unidad de descripción y nivel de descripción. El resto dependerá de la naturaleza de la unidad de descripción (Consejo Internacional de Archivos, 2000).

Los 26 elementos que componen las ISAD (G) se estructuran en 7 áreas de información descriptiva, de los cuales se utilizarán los que más interesa al objeto. Estas áreas son:

1. **Área de mención de identidad** (contiene la información esencial para identificar la unidad de descripción)
 - a. Código(s) de referencia
 - b. Título
 - c. Fecha(s) extremas de producción
 - d. Nivel de descripción
 - e. Extensión y soporte de la unidad de descripción (cantidad, volumen o tamaño)

2. **Área de contexto** (contiene información relativa al origen y custodia de la unidad de descripción)
 - a. Nombre(s) del productor (es)
 - b. Historia institucional/Reseña biográfica
 - c. Historia archivística
 - d. Forma de ingreso



3. **Área de contenido y estructura** (contiene información relativa al objeto y organización de la unidad de descripción)
 - a. Alcance y contenido
 - b. Información sobre valoración, selección y eliminación
 - c. Nuevos ingresos
 - d. Sistema de organización

 4. **Área de condiciones de acceso y utilización** (contiene información relativa a la accesibilidad de la unidad de descripción)
 - a. Condiciones de acceso
 - b. Condiciones de reproducción
 - c. Lengua/escritura del material
 - d. Características físicas y requisitos técnicos
 - e. Instrumentos de descripción

 5. **Área de documentación asociada** (contiene información relativa a aquellos documentos que tienen una relación significativa con la unidad de descripción)
 - a. Existencia y localización de los documentos originales
 - b. Existencia y localización de copias
 - c. Unidades de descripción relacionadas
 - d. Nota de publicaciones

 6. **Área de notas** (contiene información especial y aquella otra que no ha podido incluirse en ninguna de las demás áreas)
 - a. Notas

 7. **Área de control de la descripción** (contiene información relativa al cómo, cuándo y quién ha elaborado la descripción archivística)
-



- a. Nota del archivero
- b. Reglas o normas
- c. Fecha(s) de la(s) descripción(es)

Con el objetivo de proporcionar a los usuarios instrumentos para la solicitud y consulta de representaciones digitales de documentos de su interés y tomando los elementos anteriormente mencionados surgen las aplicaciones para la gestión de archivos históricos, sistemas que contribuyen considerablemente a la organización, conservación y preservación de los documentos de archivo, además de facilitar a gran escala el trabajo del archivero.

Un sistema de información puede estar constituido por cualquier producto del desarrollo informático que involucre un procesamiento de datos e información, y que apoye a las organizaciones en la optimización, mejoramiento y desarrollo de sus procesos (Martínez, 2004), por lo que los sistemas de gestión de documentos históricos constituyen un caso específico de sistemas de información.

Entonces se concluye que un sistema de recuperación de la información es el proceso donde se accede a una información previamente almacenada, mediante herramientas informáticas que permiten establecer ecuaciones de búsqueda específicas. Dicha información ha debido de ser estructurada previamente a su almacenamiento.

1.3. Relevancia y pertinencia de los resultados

El orden de aparición de los resultados de cualquier consulta, tiende generalmente a estar condicionado por criterios definidos en la implementación propia de cada sistema, en este caso estaría dado por la relevancia y pertinencia de los documentos contenidos en la aplicación.

Un documento es considerado relevante cuando satisface una determinada consulta. Una consulta es la expresión de una necesidad de información, pero ambas no son exactamente lo mismo. Un documento puede considerarse relevante si el contenido de su descripción posee alguna significación o importancia con motivo de la pregunta realizada por el usuario. Por tanto, la relevancia queda asociada con el concepto de la relación existente entre los contenidos de la descripción de un documento y los contenidos de una consulta, sin importar en ese momento el orden en que son mostrados.



Según Soergel un documento es pertinente si este es relevante y al mismo tiempo apropiado para una persona en concreto con respecto de una necesidad. El término apropiado implica que la persona pueda entender el documento y aplicar la información extraída. Por tanto, un documento puede ser relevante para una consulta y al mismo tiempo puede no ser pertinente para un usuario que no tenga los conocimientos necesarios para entenderlo, o no esté escrito en un idioma que conozca. Se trata de un concepto intrínsecamente subjetivo (Martínez, 2004).

Por todo lo anteriormente planteado se decide que el ordenamiento de los resultados no se realizará siguiendo modelos específicos, sino que se implementarán en este trabajo de diploma otros mecanismos para ordenar dichos resultados por título, productor, código de referencia, unidad y nivel de descripción, de manera que el usuario pueda definir atendiendo a cuál de estos criterios desea ordenar los resultados.

1.4. Procesos de búsqueda en sistemas de gestión de documentos históricos

1.4.1. ICA-Atom

Es un *software* creado para la empresa canadiense *Artefactual Systems* bajo patrocinio del Consejo Internacional de Archivos (ICA) y de Unesco⁴. Utiliza una capa de abstracción de bases de datos que lo hace ser compatible con PostgreSQL, SQLite, SQLServer, Oracle, pero ha sido utilizado MySQL para su desarrollo. Utiliza el lenguaje de programación PHP para gestionar las peticiones y respuestas entre los clientes web y la lógica de la aplicación (ICA Atom, 2013).

El sistema posee un buscador que permite realizar búsquedas sencillas y avanzadas. La búsqueda sencilla se realiza a través de una palabra clave o una frase y la avanzada permite ser más específico en cuanto a los criterios de búsqueda, además se puede filtrar la información por institución, tipo general de material, etc.

1.4.2. Albalá

Sistema privado de Gestión Integrado de Centros Archivísticos, desarrollado por la empresa Baratz. Está basado tecnológicamente en estándares y sistemas abiertos, diseñado para satisfacer las necesidades de todo tipo de archivos, ya sean públicos o privados, administrativos o históricos (Fernández, 2013).

⁴ Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura. Se fundó el 16 de noviembre de 1945 con el objetivo de contribuir a la paz y a la seguridad en el mundo mediante la educación, la ciencia, la cultura y las comunicaciones.



Albalá solo posee la búsqueda sencilla y la realiza por una palabra o frase. Los resultados son mostrados en forma de lista informando al usuario la fecha y una breve descripción del documento; permite al usuario hacer comentarios al respecto.

1.4.3. Archon

Es una plataforma⁵ unificada para la descripción archivística y acceso a la información, que proporciona una manera de registrar información descriptiva sobre las colecciones y objetos⁶ digitales y un medio para ver, buscar y navegar por la información publicada en su sitio público (Archives, 2013).

Este sistema solo realiza la búsqueda sencilla, pero permite filtrar la información, los resultados son mostrados poniendo primeramente el nombre de la empresa, la cantidad de páginas que contienen cada uno de los archivos encontrados y posibilita su descarga.

1.4.4. Archivo 3000

Es un software propietario para la gestión de archivos que utilizando las normas ISAD (G) e ISAAR (CPF) permite la introducción automatizada de datos, está perfectamente capacitado para cubrir las necesidades de cualquier tipo de archivo. Además, permite hacer una jerarquía del archivo aún mayor, permitiendo también adaptar y/o crear cualquier cuadro de clasificación (Archivo, 2013).

El generador de consultas permite búsquedas de los usuarios de diseño que incluyen todas las palabras, frases exactas, palabras para excluir y palabras parciales. La interfaz de resultados se divide en dos paneles, con la izquierda que muestra una vista previa de los resultados de búsqueda y la derecha que muestra el texto completo del documento seleccionado.

1.4.5. Portal Pares

Se trata de una plataforma común que integra las bases de datos de varios archivos de carácter histórico nacional. Pares ofrece un acceso libre y gratuito, a cualquier persona interesada en acceder a los documentos con imágenes digitalizadas de los Archivos Españoles. Además, implementa la norma internacional ISAD (G) (Pares, 2013).

⁵ Sistema operativo o sistemas complejos, ya sea de hardware o software, sobre el cual un programa pueda ejecutarse.

⁶ En este contexto hace referencia a un conjunto de recursos digitales de cualquier tipo de formato.



El sistema cuenta con sistemas de búsqueda sencilla y avanzada, y los resultados son mostrados diferenciando el lugar donde se encuentran, y seguidamente el nombre del documento que contiene la información deseada.

1.4.6. ArchiVenHis

Surge en el 2007 como un sistema orientado a resolver las necesidades más crecientes de almacenamiento digital de la información en el Archivo General de la Nación de la República Bolivariana de Venezuela (AGN), con las especificaciones particulares de esta institución. Su versión 1.0 se registró en el Centro Nacional de Derecho de Autor de Cuba (CENDA) en febrero 2010 con Número de Registro: 498-2010.

Dicha herramienta facilita el proceso de descripción de los fondos documentales bajo la custodia de la institución según la norma ISAD (G) y posterior recuperación utilizando las búsquedas sencilla, avanzada y exploratoria. El sistema muestra como resultado de estos procesos el nombre de los documentos cuyas descripciones contengan similares parámetros y permite su visualización o solicitud.

1.5. Análisis de las soluciones encontradas

Estos sistemas informáticos existentes en el mercado mundial y nacional presentan características comunes, entre las que se pueden mencionar la oferta de bases de datos que contienen descripciones de la documentación contenida en los archivos, a los cuales se tiene acceso a través de la introducción de mecanismos de búsqueda, utilizando para ello la combinación de palabras clave.

Luego del estudio se ha determinado que ninguna de estas variantes por independiente soluciona el problema planteado y ratifican la necesidad de incorporar al sistema de gestión de documentos históricos Dexcriba el módulo de recuperación de la información, que permita la recuperación de información que en materia de archivo se generan, conduciendo al desarrollo de una aplicación que resuelva el problema del estudio y localización de los documentos de archivo.



1.6. Tecnologías necesarias para la implementación del módulo

1.6.1. Lenguajes de programación

En cuestiones de informática la Real Academia de la Lengua Española define lenguaje, como un conjunto de signos y reglas que permite la comunicación con un ordenador (RAE, 2013).

Un lenguaje de programación es un idioma artificial diseñado para expresar instrucciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado de un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es un proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático (Mora, 2010).

Los lenguajes de programación fueron creados por el hombre con el fin de modelar y hacer en las computadoras, programas que simulen la realidad en la que se desarrolla la sociedad. Actualmente existen diferentes lenguajes de programación para desarrollar tanto aplicaciones de escritorio como aplicaciones para la web, los cuales han ido surgiendo debido a las necesidades y tendencias de las plataformas. Para la implementación del módulo se utilizarán los siguientes lenguajes:

1.6.1.1. Lenguaje del lado del servidor: PHP v5.3

PHP es un lenguaje de programación de uso general de *script* del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo (PHP, 2011).



Ventajas de PHP

- Puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux⁷, Microsoft Windows⁸, Mac OS X⁹, entre otros.
- No se encuentra limitado a resultados en HTML. Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF¹⁰ y películas Flash. También puede presentar otros resultados, como XHTML¹¹ y archivos XML. PHP puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla.
- Soporta la mayoría de servidores web actuales, incluyendo Apache, Microsoft Internet Information Server, Personal web Server, Netscape e iPlanet y muchos otros.
- La potencialidad de este lenguaje sobre las características antes mencionadas es evidente, ya sea por ser “código abierto”, multiplataforma o por el gran soporte de bases de datos con que cuenta.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, entre ellos MySQL¹² y PostgreSQL¹³.
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz.
- No requiere definición de tipos de variables.

1.6.1.2. Lenguajes del lado del cliente: *JavaScript v 1.8*

JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Fue creado por Brendan Eich en la empresa *Netscape Communications*. Tiene como características principales las siguientes:

⁷ Linux es un núcleo libre de sistema operativo basado en Unix. Es uno de los principales ejemplos de software libre y de código abierto.

⁸ Microsoft Windows es el nombre de una familia de sistemas operativos desarrollados y vendidos por Microsoft.

⁹ Mac OS X, es una serie de sistemas operativos basados en Unix desarrollado, comercializado y vendido por Apple Inc. que ha sido incluido en su gama de computadoras Macintosh desde 2002.

¹⁰ PDF (sigla del Inglés *portable document format*, formato de documento portátil) es un formato de almacenamiento de documentos digitales independiente de plataformas de software o hardware.

¹¹ XHTML, Siglas del Inglés *eXtensible HyperText Markup Language*. XHTML es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros.

¹² Sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

¹³ Servidor de bases de datos relacional orientado a objetos. Está dirigido por una comunidad de desarrolladores y organizaciones comerciales que se hacen llamar PGDG (por sus siglas en Inglés, PostgreSQL *Global Development Group*).



- Es utilizado principalmente para crear páginas web dinámicas.
- No es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia.
- No es necesario declarar los tipos de variables que van a utilizarse.
- No puede escribir automáticamente al disco duro.

JavaScript presenta disímiles ventajas dentro de las cuales se destacan que es fácil de integrar con otros lenguajes y que es compatible con la mayoría de los navegadores modernos. Su principal desventaja está en que los usuarios pueden deshabilitar *JavaScript* en su navegador (Issi, 2002).

AJAX

AJAX, es el acrónimo para *Asynchronous JavaScript + XML*, en realidad no es una tecnología sino la combinación de otras como: XHTML y CSS para crear una presentación basada en estándares, DOM¹⁴ (*Document Object Model*) para la interacción y manipulación dinámica de la presentación, XML, XSLT¹⁵ y JSON¹⁶, para el intercambio y la manipulación de información, XMLHttpRequest para el intercambio asíncrono de información y *JavaScript* para unir todas las demás tecnologías (Pérez, 2010).

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor (w3, 2010).

1.6.1.3. Lenguajes de Marcado

Un lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del mismo (Unidad, 2013). A menudo suelen ser confundidos con lenguajes de programación. Sin embargo, no son lo mismo, ya que el lenguaje de marcado no tiene funciones aritméticas o variables, como poseen los lenguajes de programación.

¹⁴ Define cómo los objetos de una página Web son representados, qué atributos tienen y cómo se manipulan. DOM es una interfaz para que aplicaciones y scripts puedan acceder a un documento pudiendo modificar su estructura y su contenido.

¹⁵ XSLT o Transformaciones XSL es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.

¹⁶ JSON, acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de *JavaScript* que no requiere el uso de XML.



A continuación son descritos dos lenguajes de marcas usados en el desarrollo del módulo propuesto: HTML y XML.

1.6.1.3.1. Lenguaje de Marcado de Hipertexto (HTML) v4.01

El Lenguaje de Marcas de Hipertexto (HTML, por siglas en inglés Hypertext Markup Language), es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertextos, que es el formato estándar de las páginas web. Fue creado por Tim Berners-Lee en 1986 y ofrece a los autores los medios para (w3c, 2012):

- Publicar documentos en línea con encabezados, textos, tablas, listas, fotos.
- Obtener información en línea a través de enlaces de hipertexto.
- Diseñar formularios para realizar transacciones con servicios remotos, para su uso en la búsqueda de información, hacer reservas y pedir productos.
- Incluir hojas de cálculo, videoclips, clips de sonido y otras aplicaciones directamente en sus documentos.

1.6.1.3.2. Lenguaje de Etiquetado Extensible (XML)

Formato simple de texto muy flexible derivado de SGML¹⁷ (ISO¹⁸ 8879). Originalmente diseñado para afrontar los retos de las grandes publicaciones electrónicas. XML es un lenguaje que permite jerarquizar y estructurar la información y describir los contenidos dentro del propio documento, así como la reutilización de partes del mismo. La información estructurada presenta varios contenidos (texto, imágenes, audio, etcétera) (w3, 2010).

Ofrece un formato para la descripción de datos estructurados, lo que facilita que las declaraciones de contenido sean más precisas y los resultados de búsquedas sean más significativos. Proporciona interoperabilidad mediante un formato basado en estándares flexibles y abiertos, con formas nuevas de acceso a las bases de datos existentes y de entregar datos a clientes de la web.

¹⁷ SGML son las siglas de *Standard Generalized Markup Language* o "Estándar de Lenguaje de Marcado Generalizado". Consiste en un sistema para la organización y etiquetado de documentos.

¹⁸ ISO (por sus siglas en inglés, *International Organization for Standardization*): es una organización de carácter voluntario fundada en 1946 que es responsable de la creación de estándares internacionales en muchas áreas, incluyendo la informática y las comunicaciones. Está formada por las organizaciones de normalización de sus países miembro.



1.6.1.4. CSS v2.1

Las Hojas de Estilo en Cascada o CSS (por sus siglas en inglés, *Cascading StyleSheets*) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas (w3c, 2012).

A continuación se describen algunas ventajas de usar CSS:

- Se obtiene mayor control de la presentación del sitio al poder tener todo el código CSS reunido en uno, lo que facilita su modificación.
- Al poder elegir el archivo CSS que se desea mostrar, puede aumentar la accesibilidad ya que permite asignarle un código CSS concreto a personas con deficiencias visuales, por ejemplo. Esto lo detecta el navegador web.
- Permite hacer mucho más legible el código HTML al tener el código CSS a parte.
- El uso del CSS permite ahorro de tiempo y trabajo al poder seguir varias técnicas (bordes redondeados, sombra en el texto, sombra en las cajas, entre otras) sin necesidad de usar un editor gráfico.

1.6.2. Servidor web: Apache v2.2

Apache es un servidor HTTP de dominio público el cual está basado en el sistema operativo Linux. Fue desarrollado en 1995 y actualmente es uno de los servidores HTTP más utilizados en la red. Está patentado por licencia BSD¹⁹. Según algunas estimaciones es utilizado para *hosting* por más del 50 % de los sitios web en todo el mundo. La versión original de Apache fue escrita para UNIX, pero nuevas versiones funcionan con OS/2, Windows y otras plataformas. Este *software* libre cuenta con una gran nombradía, precisamente porque brinda facilidades como las siguientes (Apache, 2013):

- Es capaz de funcionar en una muchedumbre de sistemas operativos, lo que lo hace prácticamente global.
- Es una tecnología gratuita de código abierto, de manera que se puede ver lo que se está instalando en el servidor sin ninguna puerta trasera, sin tener nada oculto.

¹⁹ La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público.



- Es altamente adaptable de diseño modular. Es muy simple extender las capacidades del servidor web Apache. Actualmente existen numerosos módulos para Apache que son flexibles a este, cualquier persona que tenga una notable práctica en la programación de C²⁰ o Perl²¹ puede escribir un módulo para realizar una función determinada.
- Apache utiliza una gran suma de Perl, PHP y otros lenguajes de *script*. También trabaja con Java. Teniendo todo el puntal que se necesita para tener páginas dinámicas. Además permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Es de fácil configuración en la creación y gestión de *logs* (registros). Apache permite la creación de ficheros de *log* a la medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

1.6.3. Lenguaje de modelado:

El lenguaje de modelado de objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software orientado a objetos. El uso de un lenguaje de este tipo, es más sencillo que la propia programación, pues existen menos medios para verificar de forma efectiva el correcto funcionamiento del modelo. A continuación se describen las características del lenguaje de modelado utilizado para el diseño de la solución.

1.6.3.1. Lenguaje Unificado de Modelado (UML)

Los lenguajes de modelado son un conjunto estandarizado de símbolos y sus relaciones que permiten expresar un sistema de información mediante un esquema teórico, el cual representará su diseño. El Lenguaje Unificado de Modelado o UML (por sus siglas en Inglés, *Unified Modeling Language*) es un lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas de *software*, así como para el modelado del negocio. Surgió en 1994 por iniciativa de Grady Booch y James Rumbaugh para combinar sus dos famosos métodos: el de Booch y el OMT (*Object Modeling Technique*), pero no fue hasta 1997 que fue adoptado como estándar por el OMG²². Este lenguaje de modelado es útil para resolver una gran diversidad de problemas de ingeniería, desde procesos sencillos y aplicaciones de un

²⁰ C es un lenguaje de programación creado en 1972 por Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL.

²¹ Perl es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado bourne shell, AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación.

²² OMG (por sus siglas en Inglés, *Object Management Group*): organización que promueve estándares para la industria.



solo usuario a sistemas concurrentes y distribuidos. UML permite crear los diagramas que se van generando durante el proceso de ingeniería en el desarrollo de un sistema informático (UML, 2011).

1.6.4. Framework de desarrollo.

Un *framework* es un esquema o patrón, el cual provee funcionalidades genéricas que pueden ser utilizadas con el fin de desarrollar aplicaciones de manera rápida, fácil, modular y sencilla, ahorrando tiempo y esfuerzo. También es posible que esta tecnología defina una estructura para una aplicación completa, o solo se centre en un aspecto de la misma. A continuación se describen los *frameworks* de desarrollo que son utilizados en la implementación del módulo.

1.6.4.1. CodeIgniter v2.1.3

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su principal objetivo es brindar la facilidad a los usuarios de desarrollar proyectos con mayor rapidez de lo que alcanzarían si lo tuvieran que escribir desde cero, proporcionándole una serie de librerías para tareas comúnmente necesarias como: acceder a una base de datos, enviar correos, validar datos de un formulario, mantener sesiones, manipular imágenes, entre otras (CodeIgniter, 2013)

Es importante destacar que este *framework* implementa el patrón arquitectónico Modelo-Vista-Controlador (MVC), que es un estándar de programación que puede ser utilizado tanto para la realización de sitios web como para programas de escritorio o de línea de comandos. Además, muchas de sus utilidades y modos de funcionamiento son opcionales, lo que hace que el usuario disponga de una mayor libertad a la hora de desarrollar aplicación web.

1.6.4.2. jQuery v1.8.3

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.



jQuery es *software* libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT²³ y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio, de ahí el lema que acompaña al nombre de la librería: *Write less, do more* que traducido al español significa escribe menos y haz más. Consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX (Murphey, 2010).

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

jQuery ui v1.9.2

jQuery UI es una biblioteca de componentes para el *framework* jQuery que le añaden un conjunto de *plugins*, *widgets*²⁴ y efectos visuales para la creación de aplicaciones web. Cada componente o módulo se desarrolla de acuerdo con la filosofía de jQuery (*find something, manipulate it*: encuentra algo, manipúlalo).

Es un conjunto completo de controles UI. Cada control tiene un conjunto de opciones configurables y se les pueden aplicar estilos CSS (Jquery, 2013).

1.6.5. Herramientas de programación

Actualmente existen una serie de herramientas de programación que brindan al programador cierta comodidad para desarrollar *software*. Uno de los factores a tener en cuenta en el desarrollo de dichos programas es la selección de un IDE²⁵ correcto, que además sea capaz de apoyar al desarrollador en términos como el completamiento de código y el detector de errores.

A continuación se detallan las características de Netbeans, entorno definido por los desarrolladores de Dexcriba para la implementación.

²³ Licencia cuyo objetivo es que tu código sea accesible para el mayor número de desarrolladores y trabajos derivados posible, sin excluir que el código se pueda utilizar en software propietario. El mensaje básico de esta licencia es "Eres libre para usar este código como quieras". Es compatible con la GNU GPL, y es corta, sencilla, y fácil de entender.

²⁴ En informática, un *widget* es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets*.

²⁵ Entorno de Desarrollo Integrado, llamado también IDE por sus sigla en inglés de Integrated Development Environment, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica



1.6.5.1. Entorno de desarrollo integrado (IDE): Netbeans v7.2

Permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de *software* llamados módulos. Un módulo es un archivo Java²⁶ que contiene clases java escritas en este lenguaje para interactuar con las APIs de NetBeans y un archivo especial (*manifest file*) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos.

NetBeans es un proyecto de código abierto con una gran base de usuarios, una comunidad en constante crecimiento y con más de un centenar de socios en todo el mundo. Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Tiene soporte para varios lenguajes incluyendo PHP, JavaScript, HTML y CSS (NetBeans, 2011).

Dentro de las características de NetBeans se pueden encontrar las siguientes:

- El depurado de las aplicaciones es más sencillo.
- Soporte mejorado para consumir servicios web y conectarse a bases de datos
- Autocompleta código de los lenguajes JavaScript, CSS, HTML y PHP
- Compatibilidad total con PHP 5.3.
- Es multiplataforma
- Integración con Sistemas de Control de Versiones

1.6.6. Sistema gestor de base de datos

Para C. J. Date, un Sistema de Gestión de Bases de Datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma (Date, 2001).

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

²⁶ Lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90.



Los SGDB brindan facilidad a la hora de elaborar tablas y establecer relaciones entre las informaciones contenidas en ellas. Pueden mantener la integridad de una base de datos permitiéndole a más de un usuario actualizar un registro al mismo tiempo y también puede impedir registros duplicados en una base de datos.

1.6.6.1. PostgreSQL v9.1

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es un sistema de gestión de bases de datos de código abierto que en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema (PostgreSQL, 2007).

Para esta versión se han añadido tablas *unlogged* para incrementar el rendimiento y se ha mejorado la internacionalización pudiendo especificar la regionalización de los textos por cada columna, no por tabla como se hacía hasta ahora.

1.6.7. Herramientas CASE

Acrónimo de *Computer Aided Software Engineering* (Ingeniería de *Software* Asistida por Computadoras). Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un *software*.

Estas se iniciaron con un procesador de palabras que fue usado para crear y manipular documentación. Durante los años setenta vieron la introducción de técnicas gráficas y diagramas de flujo de datos. Sobre este punto, el diseño y especificaciones en forma pictórica han sido extremadamente complejos y consumían mucho tiempo para realizar cambios. La introducción de las herramientas CASE para ayudar en este proceso ha permitido que los diagramas puedan ser fácilmente creados y modificados, mejorando la calidad de los diseños de *software* (Murillo, 1999).



1.6.7.1. Visual Paradigm v8.0

Visual Paradigm es una herramienta CASE que provee el modelado de procesos de negocio, además de un generador de mapeo de objetos relacionales para los lenguajes de programación Java y PHP. Es una herramienta que sustenta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Posibilita la importación y exportación de ficheros XML y se puede generar bases de datos. Este está diseñado para una amplia gama de usuarios, incluidos los ingenieros de *Software*, Analistas de Sistemas y Analista de Negocios. Cualquiera de los cambios que se realicen en el código existente puede reflejarse en el modelo (Pressman, 2002).

Visual Paradigm soporta un conjunto de lenguajes, tanto en la generación de código como en la ingeniería inversa. Puede generar código a partir de los modelos y viceversa. Cualquiera de los cambios que se realicen en el código existente puede reflejarse en el modelo. Esta herramienta visual permite construir la aplicación con mayor rapidez, mayor exactitud, mejor trabajo en equipo y fácil de utilizar.

Facilita la interoperabilidad con otras herramientas CASE y con la mayoría de los IDE's principalmente, permitiendo la integración de todos los componentes. Aunque es una herramienta gratuita, este programa se encuentra bajo licencias que no permiten el estudio ni modificación de la misma.

Se selecciona Visual Paradigm en su versión 8.0 ya que puede ser instalada en sistemas GNU/Linux, así como en diferentes versiones de *Windows*. Además, es la herramienta estudiada en clases por los autores de esta investigación y se puede integrar con *NetBeans*, herramienta de desarrollo seleccionada para implementar el módulo de recuperación de la información.

1.7. Metodología de desarrollo de *software*

El proceso de desarrollo de *software* requiere una metodología de desarrollo capaz de conducir este proceso, de manera que desarrolladores y clientes queden satisfechos con el producto en cuestión de manera tal que se minimicen los riesgos del fracaso.

Una metodología de desarrollo de *software* es un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de *software*, que indican paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben tener. Además, detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla (Menéndez, 2013).



1.7.1. Proceso Unificado de Desarrollo de Software (RUP)

Para el desarrollo del módulo será empleada la metodología de desarrollo RUP según lo que establece el programa de mejora que se está llevando a cabo en la Universidad con el objetivo de alcanzar una certificación internacional del nivel 2 del modelo CMMI²⁷.

El Proceso Unificado de Desarrollo, es el resultado de una convergencia entre *Rational Approach* y *Objectory*, su primera versión recibió el nombre de *Rational Objectory Process* y fue puesta en el mercado en el año 1998.

RUP unifica los mejores procesos de metodologías anteriores, es orientado a objetos y está preparado para desarrollar grandes y complejos proyectos. Se caracteriza por ser iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura. Las actividades se agrupan en grupos lógicos, definiendo 9 flujos de trabajo principales que guían el desarrollo del *software* a través de las 4 fases: Inicio, Elaboración, Construcción y Transición, que componen el ciclo de vida del proyecto. En su modelación define un grupo de elementos que determinan quién (representado por los trabajadores que definen el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo, realizan las actividades y son propietarios de elementos), qué (materializado en productos tangibles del proyecto que son producidos, modificados y usados por las actividades, pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables que en su totalidad son llamados artefactos), cómo (son tareas que tienen un propósito claro, son realizadas por los trabajadores) y cuándo (secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable) se realizarán las actividades (Jacobson, 2000).

1.8. Conclusiones parciales

En el desarrollo del capítulo fueron analizados los principales sistemas de gestión de documentos históricos nacionales e internacionales, con el propósito de estudiar y seleccionar el que más se ajustara al objetivo de la investigación. Según el estudio realizado los autores llegaron a la conclusión de que ninguno por sí solo resuelve el problema en cuestión, teniendo en cuenta los argumentos expuestos.

²⁷ Modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de administración como de ingeniería de sistemas y software



Además, fueron descritas cada una de las herramientas, tecnologías y lenguajes de programación que fueron definidos para el desarrollo del sistema en general, los cuales lógicamente se aplican al módulo, el cual será integrado a dicho sistema. Todo este proceso está guiado por la metodología RUP, mediante la cual se logra gestionar de forma eficiente el desarrollo del *software* y garantiza la calidad del producto final.



Capítulo 2. Características del módulo Recuperación de la información.

Actualmente el SGDH Dexcriba no brinda información sobre el material custodiado en él, esta capacidad constituye, en la actualidad, un aspecto de gran importancia para los SGDH debido a que contribuye considerablemente en la localización y difusión de la memoria histórica de las instituciones que lo utilizan. En el presente capítulo se realiza una propuesta para dar solución al problema anteriormente expuesto, relacionado con los procesos de búsqueda dentro del propio sistema.

2.1. Descripción de la propuesta de solución

El desarrollo de un módulo que garantice que la recuperación de la información contenida en Dexcriba, le proporciona al sistema una nueva funcionalidad que contará con subsistemas de búsqueda sencilla, avanzada y exploratoria que permitirán al usuario una localización rápida y efectiva de los documentos propios del sistema en cuestión.

2.2. Modelo de dominio

En el presente trabajo de diploma existe poca estructuración de los procesos del negocio, además no hay claridad suficiente en la identificación de procesos o actividades en el negocio haciéndose engorroso y poco exacto la descripción de los mismos, siendo así, se selecciona la realización del modelo de dominio como una de las alternativas que ofrece RUP para la comprensión del contexto.

Un modelo de dominio captura los tipos de objetos más importantes en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema (Jacobson, 2000). A través de él se definen los principales conceptos que se manejan en el dominio del sistema en relación con los usuarios y desarrolladores, así como las relaciones que se establecen entre ellos. Su elaboración posibilita a los involucrados manejar un vocabulario común lo que permite un mejor entendimiento del contexto en que se sitúa el sistema. Se representa mediante de un diagrama de clases de UML.

2.2.1. Descripción de los principales conceptos

Para una mejor comprensión del diagrama de modelo de dominio se proporciona una descripción de los conceptos fundamentales relacionados con el dominio del sistema.



Usuario: es la persona que realiza las búsquedas sencilla, avanzada o exploratoria. Además, puede exportar e importar el resultado obtenido.

Información: es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje, en este caso descripciones archivísticas contenidas por el sistema.

Criterios de búsqueda: contienen los campos obligatorios para cada una de las descripciones contenidas en el sistema, garantizando la recuperación de manera uniforme en el sistema.

Búsqueda sencilla: el usuario tiene poca información de lo que desea buscar y solo escribe una palabra clave.

Búsqueda avanzada: el usuario cuenta con mayor cantidad de datos de lo que necesita encontrar y procede a la selección de una o más palabras clave para realizar la búsqueda.

Búsqueda exploratoria: el usuario cuenta con un conocimiento de las ubicaciones físicas de los documentos y navega de manera jerárquica por los diferentes niveles del archivo, hasta encontrar lo que necesita.

2.2.2. Diagrama de clases del dominio

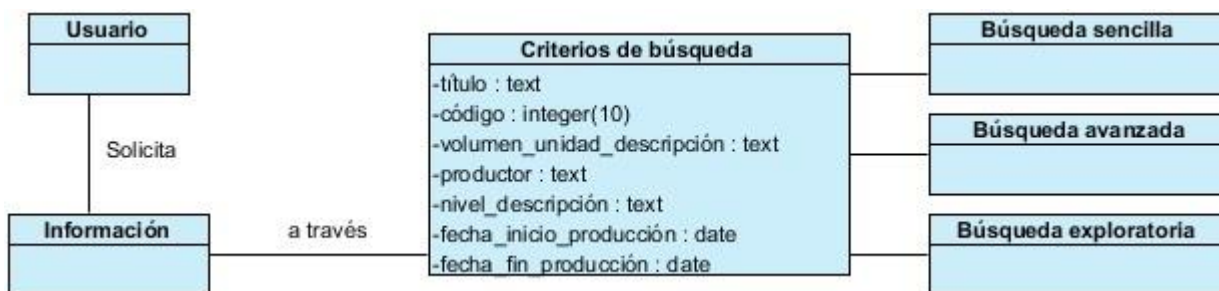


Ilustración 1: Diagrama de clases del dominio del módulo Recuperación de la Información

2.3. Especificación de los requisitos del software

Según el estándar 1233 de la IEEE: Guía para el desarrollo de Especificaciones de Requerimientos de Sistemas, un requisito se define como (IEEE, 1998):

- Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.



- Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

La obtención de estos es el proceso mediante el cual los interesados en un sistema de software descubren, revelan, articulan y entienden sus requisitos (Jacobson, 2000).

Muchas veces se hace necesario el empleo de técnicas para establecer una buena comunicación con los interesados del producto y así lograr la satisfacción del cliente.

2.3.1. Técnicas para la captura de requisitos

2.3.1.1. Análisis de sistemas existentes

Mediante el análisis de sistemas existentes es posible estudiar aplicaciones similares a la que se necesita obtener. Una vez que se tiene la concepción del funcionamiento de un *software* similar en cuanto a funcionalidades y características es más sencillo identificar los requisitos del sistema que se necesita implementar.

Para el uso de esta técnica se realizó un estudio de aplicaciones similares a la que se necesita obtener, observándose los diseños de sus interfaces, las funcionalidades que ofrecen y el grado de dificultad a la hora de interactuar con la aplicación.

2.3.1.2. Juego de roles

Variante en la que el desarrollador, el analista y cada uno de los miembros del equipo de desarrollo del *software* toman el lugar del interesado y ejecutan la actividad de trabajo que este desempeña.

En este caso los involucrados fueron los desarrolladores de cada uno de los módulos que completarán el sistema Dexcriba, con la idea de experimentar las inexactitudes y problemas ligados con cada una de las áreas por individual, con el objetivo de suministrarle al analista una perspectiva nueva del problema que le permita la obtención de los requisitos del sistema por construir.

2.3.1.3. Entrevista

Es la más tradicional de las técnicas de obtención y consiste en reuniones analista-interesado en las cuales se suceden preguntas y respuestas para extraer el dominio de la aplicación. Las preguntas que guiaron la entrevista no estructurada se encuentran en el Anexo 1.



Fueron entrevistados 3 personas con más de 5 años de experiencia en el área de la gestión documental y archivística, que han trabajado en el desarrollo de sistemas homólogos de gestión de documentos históricos como Archivenhis, entre ellos la Msc. Laritza Cabrera Barroso, analista principal del proyecto, y los ingenieros Leodan de los Ángeles Buduen y Yoani Santana Leiva, encargados de guiar el proceso de desarrollo del sistema.

La entrevista realizada permitió esclarecer las necesidades reales del módulo, así como validar la propuesta de los desarrolladores del presente trabajo de diploma sobre los criterios necesarios para la recuperación de la información, la inclusión de nuevas formas para mostrar y ordenar los resultados, y la posibilidad de exportar e importar resultados de los procesos de búsqueda.

2.3.2. Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Identifican lo que el sistema debe hacer y se mantienen estáticos sin importar con qué propiedades o cualidades estén relacionadas (Pressman, 2001). Con el desarrollo del módulo el SGDH debe ser capaz de:

- ✓ **RF1. Realizar búsqueda sencilla:** permite realizar búsquedas sencillas sobre los documentos indexados en los repositorios²⁸.
- ✓ **RF2. Realizar búsqueda avanzada:** permite realizar búsquedas avanzadas (título, repositorio, clasificación) sobre los documentos indexados de los repositorios.
- ✓ **RF3. Realizar búsqueda exploratoria:** el sistema permite encontrar los documentos a partir de una navegación por niveles simulando la estructura jerárquica del archivo.
- ✓ **RF4. Mostrar los resultados:** el sistema muestra los resultados de la búsqueda realizada.
- ✓ **RF5. Visualizar descripciones:** el sistema muestra una breve descripción del documento.
- ✓ **RF6. Ordenar resultados por título:** el sistema permite ordenar de manera ascendente o descendente los resultados obtenidos, a partir del campo "título".
- ✓ **RF7. Ordenar resultados por código:** el sistema permite ordenar creciente o decrecientemente los resultados obtenidos, a partir del campo "código de referencia".
- ✓ **RF8. Ordenar resultados por productor:** el sistema permite los resultados obtenidos, a partir del productor.

²⁸ Se refiere a un "proveedor de datos" y de forma genérica se puede definir como un lugar central donde se registran datos para su almacenamiento y conservación con propósitos diversos de seguridad o consulta posterior.



- ✓ **RF9. Ordenar resultados por volumen de la unidad de descripción:** el sistema permite ordenar creciente o decrecientemente los resultados obtenidos, a partir del volumen de la unidad de descripción.
- ✓ **RF10. Ordenar resultados de acuerdo con nivel de descripción:** el sistema permite organizar los resultados obtenidos, a partir del nivel de descripción.
- ✓ **RF11. Exportar los resultados de la búsqueda:** permite exportar el resultado filtrado y ordenado de la búsqueda realizada.
- ✓ **RF12. Importar resultados de búsqueda:** permite importar el resultado de una búsqueda que fue exportada con anterioridad.

2.3.3. Requisitos no funcionales

Los requisitos no funcionales definen propiedades y restricciones del sistema, que pueden estar relacionadas con el entorno, implementación, rendimiento, dependencia de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad (Jacobson, 2000).

Usabilidad

- ✓ **RNF1.** El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.

Eficiencia

- ✓ **RNF2.** El sistema web debe garantizar la gestión de la información, y poseer un tiempo de respuesta por transacción de hasta 5 segundos en dependencia de la colección documental, el número de usuarios realizando peticiones y el servidor donde se encuentre instalado.

Soporte

- ✓ **RNF3.** Se debe implementar el sistema siguiendo el estándar de codificación definido en el segundo capítulo.
- ✓ **RNF4.** Los componentes de *software* que integran la solución se organizarán de forma modular.

Interfaz

- ✓ **RNF5.** La interfaz gráfica del módulo debe concebirse con un ambiente sencillo y de navegación fácil e intuitiva para los usuarios.



Restricciones de diseño

- ✓ **RNF6.** Lenguaje de programación: PHP 5.3.
- ✓ **RNF7.** *Framework* de desarrollo: CodeIgniter 2.1.3.
- ✓ **RNF8.** Bibliotecas de clases: jQuery 1.8.3.
- ✓ **RNF9.** Bibliotecas de clases: jQuery ui 1.9.2.
- ✓ **RNF10.** Gestor de base de datos PostgreSQL 9.1.
- ✓ **RNF11.** Herramienta de programación Netbeans 7.2.
- ✓ **RNF12.** Diseñar un sistema compuesto por subsistemas que agrupen funcionalidades.
- ✓ **RNF13.** La interfaz web del módulo deberá implementarse haciendo uso de HTML, CSS, JavaScript y JQuery.

Software

- ✓ **RNF14.** Instalar en el servidor de bases de datos el SGBD PostgreSQL 9.1 o superior.
- ✓ **RNF15.** Instalar en el servidor web, Apache 2.2 o superior configurado adecuadamente para trabajar con CodeIgniter.
- ✓ **RNF16.** Navegador web en los clientes: Internet Explorer 9 o superior, Mozilla Firefox 11.0 o superior, Google Chrome 17.0 o superior.

2.4. Modelo de casos de uso

El modelo de casos de uso del sistema permite que los desarrolladores del *software* y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Describe lo que hace el sistema para cada tipo de usuario (Brito, 2009).

2.4.1. Actores

Los actores del sistema representan el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema y pueden intercambiar información con él o ser recipientes pasivos de información. En la Tabla 1 se define dicho actor.



| Actor | Descripción |
|---------|--|
| Usuario | Es la persona que interactúa con el sistema para realizar las búsquedas sencillas, avanzadas y exploratorias, además de gestionar el trabajo con los resultados obtenidos. |

Tabla 1: Descripción del actor del sistema

2.4.2. Diagrama de casos de uso del sistema



Ilustración 2: Diagrama de CUS

2.4.3. Validación de requerimientos: Matriz de trazabilidad

En este epígrafe se muestra mediante una matriz de trazabilidad cómo se recogen todos los requisitos funcionales en los casos de uso del sistema (CUS) identificados.

| X | RF1 | RF2 | RF3 | RF4 | RF5 | RF6 | RF7 | RF8 | RF9 | RF10 | RF11 | RF12 |
|-------------------------------------|-----|-----|-----|-----|-----|--|-----|-----|-----|------|------|------|
| CU1 | X | | | X | X | | | | | | | |
| CU2 | | X | | X | X | | | | | | | |
| CU3 | | | X | X | X | | | | | | | |
| CU4 | | | | | | X | X | X | X | X | | |
| CU5 | | | | | | | | | | | X | |
| CU6 | | | | | | | | | | | | X |
| CU1: Realizar Búsqueda Sencilla | | | | | | CU4: Ordenar resultados de las búsquedas | | | | | | |
| CU2: Realizar Búsqueda Avanzada | | | | | | CU5: Exportar resultados de la búsqueda | | | | | | |
| CU3: Realizar Búsqueda Exploratoria | | | | | | CU6: Importar resultados de la búsqueda | | | | | | |

Tabla 2: Matriz de trazabilidad



Como se puede observar en la Tabla 2 cada requisito identificado ha sido cubierto por al menos un caso de uso del sistema, lo que significa que la especificación de caso de uso propuesta satisface todas las necesidades del cliente.

2.4.4. Descripción de los casos de uso

A continuación se ofrece una descripción del CUS Realizar búsqueda avanzada. Para ver una descripción de los restantes casos de usos consultar el Anexo 2 de este trabajo.

| | | |
|---|--|--|
| Caso de Uso: | CU2 Realizar Búsqueda Avanzada | |
| Actores: | Usuario (Inicia). | |
| Resumen: | El caso de uso inicia cuando el actor procede a realizar una búsqueda avanzada sobre las descripciones de los documentos contenidos en el sistema, filtrando la búsqueda de acuerdo con seis parámetros obligatorios establecidos por la norma ISAD (G), y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso. | |
| Precondiciones: | | |
| Referencias | RF2,RF4,RF5 | |
| Prioridad | Crítica | |
| Flujo Normal de Eventos | | |
| Acción del Actor | Respuesta del Negocio | |
| 1-Introduce los parámetros de búsqueda. | 1.1 Busca en el sistema las descripciones de los documentos cuyos datos concuerden con los introducidos y muestra todos los posibles documentos debajo de los campos de búsqueda. | |
| Flujos Alternos | | |
| Línea 1.1: El sistema no encuentra ninguna descripción cuyos datos coincidan con los parámetros introducidos. | 1.1 El sistema informa que no existe ningún documento descrito con esos datos | |
| Línea 1.2: No introduce ningún dato en los criterios de búsquedas | 1.2- No se activa el botón Buscar. | |
| Línea 1.3: No presiona el botón Buscar. | 1.3- Queda cancelada la opción de búsqueda. | |
| Poscondiciones | Se muestra una lista de documentos en cuyas descripciones los valores de los campos coincidan con los criterios de búsqueda proporcionados | |

Tabla 3: Descripción del CUS Realizar búsqueda avanzada

2.5. Análisis del sistema

Durante el análisis, se analizan los requisitos que se obtuvieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo la arquitectura (Brito, 2009). A continuación se nombran los tipos de clases que son utilizados en el modelo de análisis:



CI_ [Nombre de la Clase]: la clase interfaz representa las formas de interacción entre los actores y el sistema, ventanas, formularios y comunicación con otros sistemas.

CC_ [Nombre de la Clase]: la clase controladora es la encargada de coordinar y controlar los procesos para satisfacer las necesidades surgidas a partir del levantamiento de requisitos. En ellas se realiza el procesamiento más complejo dentro del sistema.

CE_ [Nombre de la Clase]: la clase entidad modela la información almacenada en el sistema.

2.5.1. Diagramas de clases del análisis.

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa el funcionamiento del mundo real, no de la implementación automatizada del mismo. A continuación se muestra el diagrama de clase del análisis del CUS Realizar Búsqueda Sencilla, los demás pueden ser consultados en Anexo 3.

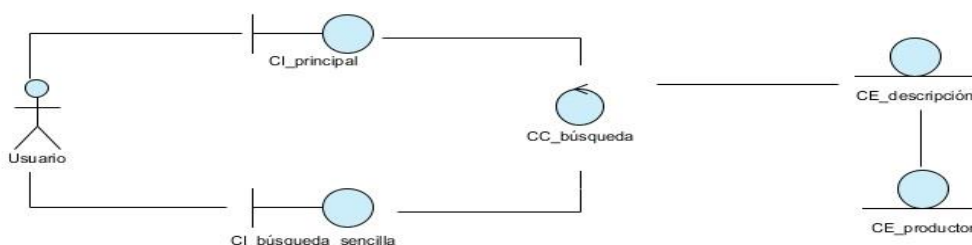


Ilustración 3: Diagrama de clases del análisis del CUS Realizar búsqueda sencilla

2.5.2. Diagramas de interacción

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

A continuación se muestra el diagrama de interacción del CUS Realizar Búsqueda Sencilla, el resto de los diagramas pueden ser consultados en el Anexo 4 de este trabajo de diploma.



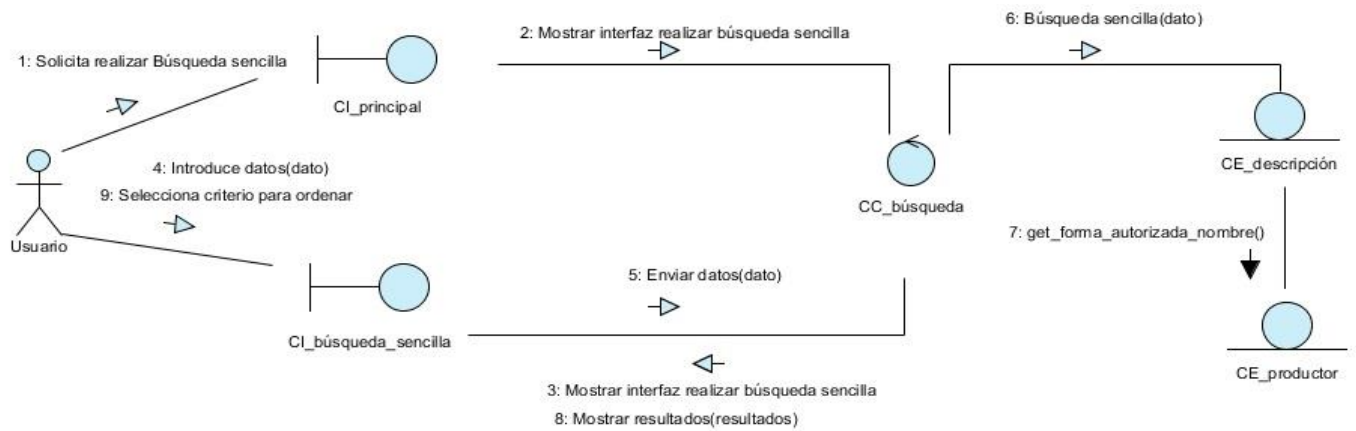


Ilustración 4: Diagrama de interacción del CUS Realizar búsqueda sencilla

2.6. Diseño del sistema

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y para documentar el diseño del sistema de *software*. Es un producto de trabajo integral y compuesto que abarca todas las clases de diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos (Pressman, 2001)

2.6.1. Diagramas de clases del diseño

Un diagrama de clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Sirve para visualizar las relaciones entre las clases involucradas en el sistema, las cuales pueden ser asociativas, de herencia, de uso y de convencimiento.

A continuación se muestra el diagrama de clases del diseño del módulo Recuperación de la información.



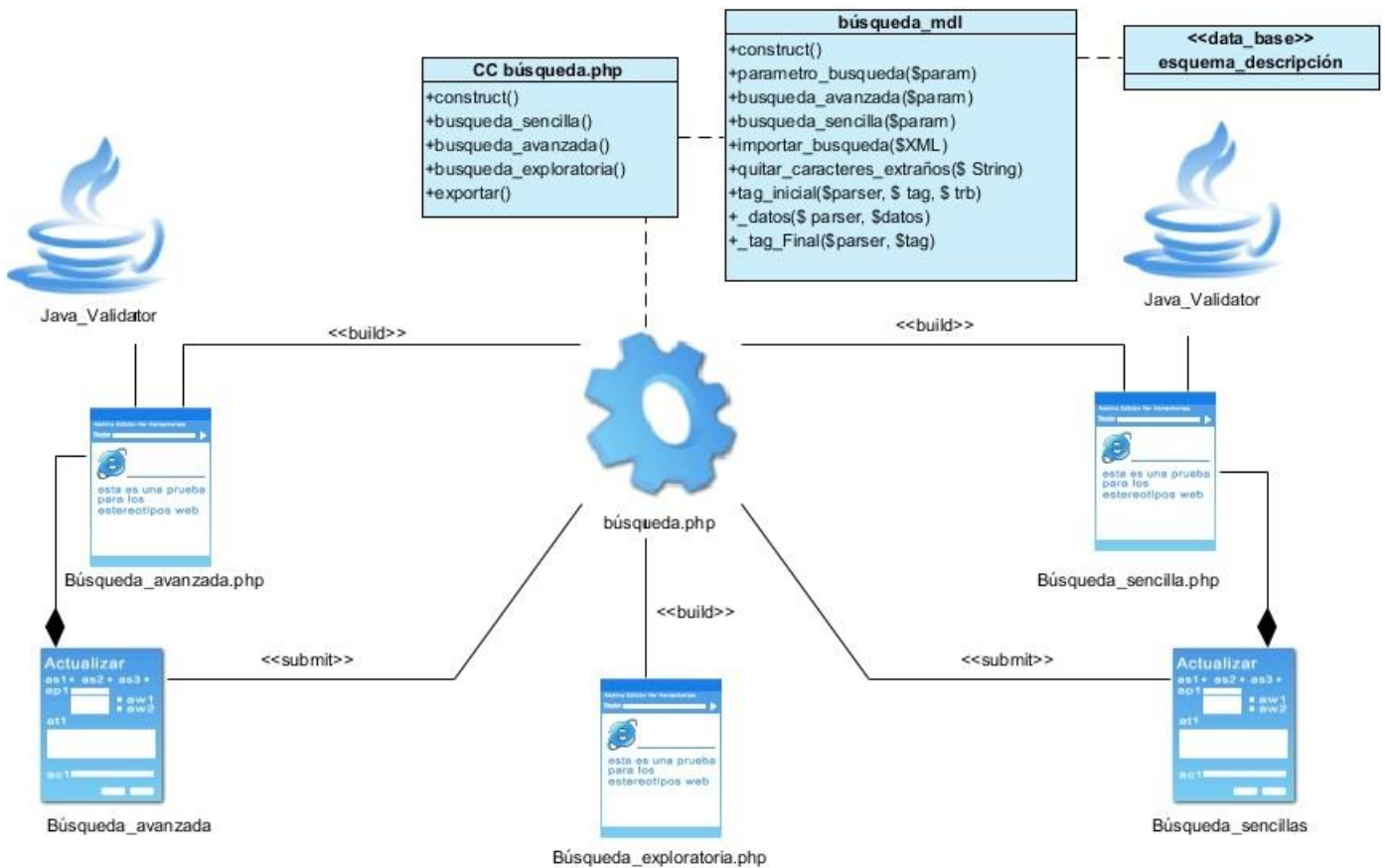


Ilustración 5: Diagrama de clases del diseño del módulo Recuperación de la información

2.7. Diseño de la base de datos

El módulo trabaja con la base de datos que posee Dexcriba. Los datos de los documentos obtenidos en el proceso de búsquedas se obtienen directamente de dicha base de datos, por lo que no será necesario construir una, ni incorporar tablas nuevas.

2.7.1. Modelo de datos

Un modelo de datos no es más que la colección de herramientas conceptuales para describir un fenómeno de la realidad objetiva a través de los objetos, sus propiedades y las restricciones de consistencia que se establecen entre ellos (Mato, 1999).

En el Anexo 5 se muestran las tablas que fueron utilizadas del esquema Descripción de la base de datos con la descripción correspondiente de las entidades necesarias para el módulo.



2.8. Arquitectura

La arquitectura de *software* es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución. La selección de un patrón de arquitectura es por lo tanto una decisión fundamental al desarrollar un sistema de *software* (Alonso, 2010).

Se propone que el sistema sea implementado sobre el *framework* CodeIgniter, el cual se basa en el patrón arquitectónico Modelo – Vista – Controlador, por lo que el módulo a desarrollar debe ajustarse a este patrón.

2.8.1. Modelo-Vista-Controlador (MVC)

MVC es un patrón de arquitectura de *software* encargado de separar la lógica de negocio de la interfaz del usuario y es más utilizado en aplicaciones web, ya que facilita la funcionalidad, el mantenimiento y la escalabilidad del sistema, de forma simple y sencilla (Bahit, 2011).

MVC divide las aplicaciones en tres niveles de abstracción:

- **Modelo:** representa la lógica de negocios. Es el encargado de acceder de forma directa a los datos actuando como intermediario con la base de datos.
- **Vista:** es la encargada de mostrar la información al usuario de forma gráfica y humanamente legible.
- **Controlador:** es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que esta, los presente al usuario, de forma humanamente legible.

Funcionamiento del patrón Modelo-Vista-Controlador

- El usuario realiza una petición.
- El controlador captura el evento(puede hacerlo mediante un manejador de eventos – *handler*-, por ejemplo).
- Hace la llamada al modelo o los modelos correspondientes (por ejemplo, mediante una llamada de retorno – *callback* -) efectuando las modificaciones pertinentes sobre el modelo.



- El modelo será el encargado de interactuar con la base de datos, ya sea en forma directa, con una capa de abstracción para ello, un *web service*, etc. y retornará esta información al controlador
- El controlador recibe la información y la envía a la vista
- La vista, procesa esta información creando una capa de abstracción para la lógica (quien se encargará de procesar los datos) y otra para el diseño de la interfaz gráfica o GUI (por sus siglas en Inglés).

La lógica de la vista, una vez procesados los datos, los acomodará en función del diseño de la GUI - *layout*– y los entregará al usuario de forma que este pueda comprenderlo

2.8.2. Patrones de diseño

Un Patrón de diseño define un esquema de refinamiento de los subsistemas o componentes dentro de un sistema, o las relaciones entre estos, describe además una estructura común y recurrente de componentes interrelacionados, que resuelve un problema general de diseño dentro de un contexto particular (Marquina, 2008).

Patrones GRASP

En la solución propuesta se evidencian varios patrones de diseño pertenecientes al grupo GRASP, acrónimo de *General Responsibility Assignment Software Patterns* (patrones generales de *software* para asignar responsabilidades). Estos patrones describen los principios fundamentales de la asignación de responsabilidades a objetos (Larman, 1999).

A continuación se describen el uso de estos en el módulo.

- **Experto:** es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele ser útil en el diseño orientado a objetos.

Ejemplo: la aplicación de este patrón se evidencia en las clases `jquery.cookie`, `jquery.hotkeys` y `jquery.jstree` que contienen la información necesaria para construir el árbol, acción que también pudiera ser realizada por la clase controladora `busqueda`, pero esta clase solo posee los datos que son necesarios para llamar al método `busqueda_exploratoria()` que es quien realmente construye la estructura en forma de árbol.



Beneficios: se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento.

- **Creador:** la creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos.

Ejemplo: la aplicación de este patrón se evidencia en la clase controladora "busqueda", donde a través del objeto *load* se cargan los elementos del marco de trabajo, dígame bases de datos, ficheros de idioma, librerías, modelos y vistas.

Beneficios: propicia encontrar un creador que se debe conectar con el objeto producido en cualquier evento.

- **Controlador:** asignar la responsabilidad del manejo de los eventos de un sistema a una clase.

Ejemplo: el empleo de este patrón se evidencia en la clase controladora "busqueda", que maneja y controla los datos obtenidos en la clase modelo "busqueda.mdl", permitiendo mostrar a los usuarios a través de las clases vistas ("*busqueda_sencilla.php*", "*busqueda_avanzada.php*" y "*busqueda_exploratoria.php*") los resultados de los diferentes procesos de búsqueda.

Beneficios: mayor potencial de los componentes reutilizables, ya que garantiza que los procesos de dominio sean manejados por la capa de aplicación y no por la de interfaz, además de tener un mayor control.

- **Bajo acoplamiento:** asignar una responsabilidad para mantener bajo acoplamiento²⁹.

Ejemplo: la utilización de este patrón se evidencia en el uso de una librería específica para la construcción de cada una de las vistas, lo que evidencia la poca dependencia entre las clases.

Beneficios: las clases no se afectan por cambios de otros componentes, además son fáciles de entender por separado y fáciles de reutilizar.

²⁹ Es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo acoplamiento no depende de muchas otras clases.



- **Alta cohesión:** asignar una responsabilidad de modo que la cohesión³⁰ siga siendo alta. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

Ejemplo: por la estrecha relación que existe entre este patrón y el de Bajo Acoplamiento se toma como ejemplo el mismo planteado para el anterior patrón.

Beneficios: mejoran la claridad y la facilidad con que se entiende el diseño. La ventaja de una gran funcionalidad soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico.

La propia implementación de CodeIgniter contiene estos dos últimos patrones nivelados pues permite el uso de los componentes de forma individual.

Patrones GOF

GoF, acrónimo de *Gang o Four*, Pandilla de los Cuatro, en su traducción al español. Los patrones de diseño, conocidos como GoF se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (Gamma, 2011). A continuación son detallados los que han sido utilizados en el diseño de la solución.

- Instancia única (*Singleton*): es un patrón de diseño que se basa en garantizar que una clase solo tenga una única instancia y proporciona un punto de acceso global a ella. CodeIgniter logra esto a través del `$this`, la utilización de este comando garantiza la instanciación de la clase una sola vez.

Ejemplo: la aplicación de este patrón se evidencia al utilizar el constructor `$this` para llamar desde las funciones a cualquiera de las funciones habilitadas en el CodeIgniter, esta llamada puede hacerse directamente desde la clase controladora "busqueda", la modelo "busqueda_md" o cualquiera de las vistas.

También y más concretamente se manifiesta la utilización de este patrón en la creación de librerías y *helpers* para las cuales, a través de la función, `get_instance()` se obtiene una copia del único constructor que instancia a todo el *framework*, esto es también visible cuando desde una vista se quiere llamar una librería o una modelo.

³⁰ Es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme



2.9. Manejo de errores

Los errores son tratados en el sistema de acuerdo con la fuente que los produce, los errores de lógica en el servidor son controlados a través del lanzamiento y captura de los errores de los mismos, mostrando la información correspondiente al usuario. En la parte del cliente los datos serán validados usando JavaScript interpretado por el navegador, dichos datos también se validan en el servidor.

2.10. Estándar de codificación

Con el propósito de hacer el código uniforme y mejorar el rendimiento de la aplicación, se determinó cumplir con los parámetros que establece el estándar de codificación propuesto por el proyecto. Dichos estándares no son más que un conjunto de reglas específicas aplicadas a un lenguaje, que reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Es notable destacar que los estándares de codificación no destapan problemas existentes, más bien evitan que los errores ocurran (Reyes, 2003).

Identación, llaves de apertura y cierre, tamaño de las líneas

Usar una indentación sin tabulaciones, con un equivalente a cuatro espacios. El uso de las llaves `{}` es seguido del nombre del método. La longitud de las líneas de código es aproximadamente de 75-80 caracteres para mantener la legibilidad del código.

Variables

Las variables deben contener solo letras minúsculas, usando el subrayado como separador en caso de ser nombres compuestos, y se llamarán razonables para indicar su propósito y contenido. Las variables muy cortas, no de palabras, solo deben utilizarse como iteradores en *bucles for* (`()`) se rigen por la nomenclatura *camelCase*.

Clases

Los nombres de las clases deben tener siempre la primera letra mayúscula, y el método constructor debe coincidir exactamente. Varias palabras deben estar separadas por un guion y no *camelCase*. Todos los otros métodos de la clase deben ser enteramente en minúsculas y el nombre que indique claramente su función, que incluye preferentemente un verbo.



Funciones

Se rigen por la nomenclatura *camelCase*. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

Estructuras de control

Se incluye *if*, *for*, *foreach*, *while*, *switch*, entre las estructuras de control y los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

2.11. Conclusiones parciales

Hasta el momento, la investigación realizada ha llegado al punto donde la utilización de RUP ha guiado satisfactoriamente el desarrollo del módulo, encontrándose listo para la implementación. Se siguió un conjunto de pasos para obtener una concepción del sistema que cumpla con los objetivos y expectativas marcados, de forma tal que se incluyen en esta sección los artefactos generados en correspondencia con los flujos de trabajo desarrollados.



Capítulo 3. Implementación y pruebas del módulo Recuperación de la información.

En el presente capítulo quedan reflejados los principales elementos referentes a la implementación del módulo, haciendo una representación de la implementación de las clases del diseño en términos de componentes y cómo estos se organizan de acuerdo con los nodos específicos en el modelo de despliegue. Se incluye, además, todo lo referente al resultado de un conjunto de pruebas realizadas al *software* para verificar su funcionamiento y cómo es capaz de responder a los requerimientos para el cual fue concebido.

3.1. Diagrama de componentes

Los diagramas de componentes ilustran las piezas del *software* que conforman un sistema. Muestran los elementos de diseño de un sistema de *software*, permitiendo visualizar con más facilidad su estructura general y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces (Jacobson, 2000).

Los componentes representan todos los tipos de elementos *software* que se incluyen en la fabricación de aplicaciones informáticas, por tanto, pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros.

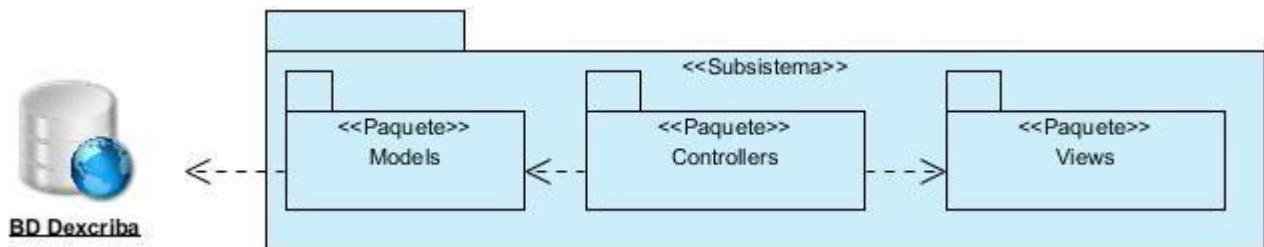


Ilustración 6: Diagrama de componentes global del módulo Recuperación de la información

En el paquete *Views* se agrupan los componentes necesarios para la interacción del usuario con el sistema, los cuales son manejados por el paquete de *Controllers*.



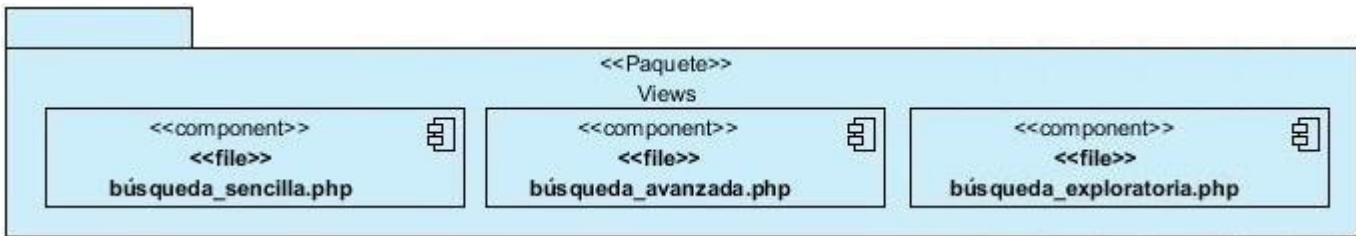


Ilustración 7: Paquete de Vistas

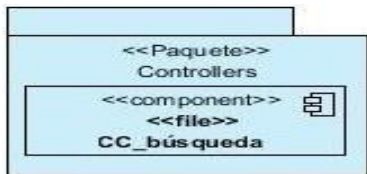


Ilustración 8: Paquete de Controladoras

El paquete *Controllers* es el rector de las actividades de la aplicación, este contiene los ficheros de código fuente que interactúan con los demás paquetes, coordinando las acciones del *software*.

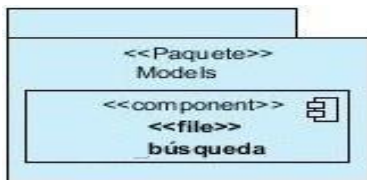


Ilustración 9: Paquete de Modelos

El paquete *Models* es el responsable de interactuar con la capa de almacenamiento de datos, para de esta forma gestionar la información con la que trabaja el sistema.

3.2. Modelo de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados mediante enlaces de comunicación (García, 2008). Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. A continuación se describen los elementos que componen el diagrama de despliegue correspondiente al módulo proveedor de datos para el SGDh.

- **Nodo PC cliente:** se refiere a las computadoras que utilizarán los usuarios para interactuar con la aplicación, en este caso se trata del proveedor de servicios. Se comunica con el Servidor de Aplicación a través del protocolo HTTP³¹.
- **Nodo Servidor web:** representa al servidor Apache donde se encontrará instalado el sistema.

³¹ HTTP (por sus siglas en inglés, Hypertext Transfer Protocol): Es un protocolo de nivel de aplicación para sistemas distribuidos. HTTP ha estado en uso por la iniciativa de la World Wide Web mundial de la información desde 1990.



- **Nodo Servidor de base de datos:** servidor PostgreSQL donde se almacenarán los datos relacionados con el sistema.



Ilustración 10: Modelo de despliegue

3.3. Pruebas

Persiguiendo el propósito de verificar la calidad de un software, se realizan un conjunto de pruebas que demuestran si las funcionalidades cumplen con lo especificado por el cliente. Para probar cualquier producto de *software* son considerados varios enfoques, donde conociendo la función específica para la que fue diseñada el producto, se pueden llevar a cabo pruebas dirigidas a demostrar que cada funcionalidad es completamente operativa, a medida que se prueban las mismas; otro enfoque sería cuando conociendo el funcionamiento del *software*, se puedan desarrollar pruebas que aseguren que todos los componentes internos se han comprobado de forma adecuada, lo cual se logra realizando pruebas a la interfaz del producto y un tercero estaría dado por las pruebas para ver la reacción del sistema ante muchas peticiones simultáneas. Estos enfoques son denominados prueba de Caja Blanca, Caja Negra y Pruebas de carga y estrés respectivamente (Jacobson, 2000).

A continuación se describen los métodos que serán utilizados en el desarrollo de las pruebas para el módulo en cuestión

3.3.1. Pruebas de Caja Blanca

La prueba de Caja Blanca es un método para el diseño de casos de prueba que está dirigido a las funcionalidades internas de un sistema. Entre las técnicas que se pueden aplicar dentro de este método se encuentra la técnica del camino básico, la cual permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico, garantizan que durante la prueba se ejecuta al menos una vez cada sentencia del programa (Pressman, 2001).

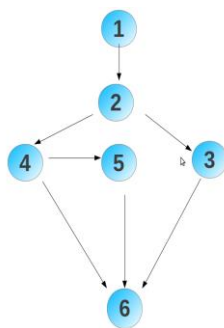
A continuación se describe cómo se realizó el diseño de los casos de prueba.



3.3.1.1. Diseño de casos de prueba de caja blanca

El caso de prueba especifica la forma de probar un sistema incluyendo las entradas, salidas y resultados esperados, así como en qué condiciones debe probarse el *software*. Para probar la calidad del módulo, fue realizada una primera etapa de pruebas correspondiente a la iteración inicial en el cual se contemplan los casos de usos críticos dentro de la aplicación, donde fueron identificadas tres no conformidades, dos de ellas asociadas a errores de validación y una vinculada con errores de sintaxis en la búsqueda avanzada, las cuales fueron solucionadas de forma satisfactoria. Las pruebas aplicadas a cada una de las funcionalidades se realizaron mediante la técnica del camino básico, la cual permitió obtener una medida de la complejidad lógica de un diseño procedimental y usarla como guía para la definición de un conjunto básico de caminos de ejecución, a los cuales se le realizaron los casos de prueba. Para aplicar esta técnica primeramente se enumera el código del método a probar como se muestra en el Anexo 6.

A partir de dicha enumeración y teniendo en cuenta la notación para las instrucciones, se construye el grafo de flujo (ver Ilustración 11).



Luego se calcula la complejidad ciclomática mediante alguna de las siguientes fórmulas:

$$V(G) = \text{Aristas} - \text{Nodos} + 2 = 7 - 6 + 2 = 3$$

$$V(G) = \text{Nodos predicados} + 1 = 2 + 1 = 3$$

$$V(G) = \text{Número de regiones} = 3$$

Ilustración 11: Grafo de flujo

El cálculo efectuado mediante las tres fórmulas ha dado como resultado el mismo valor, por lo que se puede afirmar que la complejidad ciclomática del código es 3, lo cual acota el límite superior para el número de caminos independientes que componen el conjunto básico y, consecuentemente, un valor límite superior para el número de pruebas que se deben diseñar y ejecutar para garantizar que se cubran todas las sentencias de los procedimientos.

Este número también se corresponde con el valor mínimo de casos de pruebas que deben realizarse teniendo en cuenta el procedimiento escogido. A continuación se realizan los casos de pruebas por cada uno de los caminos identificados.



| Camino | Secuencia de nodos |
|-------------------|--------------------|
| Camino básico # 1 | 1--2--4--6 |
| Camino básico # 2 | 1--2--3-- 6 |
| Camino básico # 3 | 1--2--4-- 5--6 |

Tabla 4: Caminos identificados para las pruebas de caja blanca

| Caso de prueba para el camino básico # 1 | |
|---|---|
| Nombre de la persona que realiza la prueba: María I. González del Monte | Estado de evaluación: Satisfactorio |
| Descripción | En este caso el usuario no va a buscar información ni importar una búsqueda realizada con anterioridad. |
| Entradas | |
| Condición de ejecución | |
| Resultados esperados | Se muestra la interfaz. |

Tabla 5: Caso de prueba para el camino básico #1

| Caso de prueba para el camino básico # 2 | |
|---|--|
| Nombre de la persona que realiza la prueba: María I. González del Monte | Estado de evaluación: Satisfactorio |
| Descripción | En este caso se hace la entrada de los parámetros para la búsqueda en el sistema. |
| Entradas | Algunos de los siguientes parámetros criterio, título, código, fecha de inicio, fecha fin, productor, volumen de unidad de descripción y nivel de descripción. |
| Condición de ejecución | El usuario tiene que entrar al menos un parámetro. |
| Resultados esperados | Se espera obtener la lista documentos que cumplan con los parámetros usados. |

Tabla 6: : Caso de prueba para el camino básico #2

| Caso de prueba para el camino básico # 3 | |
|--|---|
| Nombre de la persona que realiza la prueba: María I. González del Monte | Estado de evaluación: Satisfactorio |
| Descripción | En este caso el usuario importa una búsqueda realizada con anterioridad. |
| Entradas | Un xml con el resultado de la búsqueda guardada. |
| Condición de ejecución | El usuario debe tener guardado el resultado obtenido de la búsqueda realizada en el sistema con anterioridad. |
| Resultados esperados | Se espera obtener la lista de documentos. |

Tabla 7: : Caso de prueba para el camino básico #3



3.3.2. Pruebas de Caja Negra

Las pruebas de caja negra, también denominadas prueba de comportamiento, se centran en los requisitos funcionales del *software*. Esta prueba intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos, errores de rendimiento y errores de inicialización y de terminación.

Existen diversos métodos de prueba de caja negra, pero uno de los más usados es la partición equivalente, este método divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar (Pressman, 2001).

Para garantizar la calidad del sistema se aplicaron pruebas de caja de negra, diseñando 6 casos de prueba basados en los casos de uso definidos, cada uno de ellos con sus respectivas secciones y escenarios.

Para cada uno de los casos de uso planteados, a través de la matriz de datos, para comprobar que la búsqueda avanzada funcione de forma correcta, se presentan a continuación:

- ✓ V: indica válido
- ✓ I: indica inválido
- ✓ NA: que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante
- ✓ VX: representan valores de entrada de datos.
- ✓ Las variables V1, V2, V3, V4, V5, V6 y V7 significan los valores de entrada de datos para los casos de pruebas.

3.3.2.1. Diseño de casos de prueba de caja negra

Se realizaron casos de prueba a seis casos de uso documentados en el trabajo, los cuales fueron probados para garantizar el correcto funcionamiento del módulo; las pruebas realizadas se llevaron a cabo en tres iteraciones, en la primera iteración se detectaron 12 no conformidades de las cuales 7 fueron bajas (no significativas), 4 altas (significativa) y una recomendación, en la segunda iteración fue detectada una



no conformidad, y se realizó una recomendación por parte del tribunal, y finalmente en la tercera iteración no se detectaron no conformidades.

A continuación se muestra el diseño de prueba de caja negra para el CUS Realizar búsqueda sencilla, los restantes, pueden ser consultados en el Anexo 7.

| Escenario | Descripción | Buscar | Respuesta del sistema | Flujo central |
|---|---|----------------|--|---|
| EC 1.1 Realizar Búsqueda Sencilla llenado el campo y obteniendo resultados. | El sistema busca las descripciones cuyos datos concuerden con el introducido y muestra todos los posibles resultados debajo de los parámetros | V (martí) | El sistema busca todas las coincidencias con la palabra y muestra el resultado | 1-Hacer clic en el vínculo Búsqueda Sencilla en el menú principal. 2- El usuario introduce el dato y presiona el botón Buscar. 3- El sistema muestra los resultados |
| EC 1.2 Realizar Búsqueda Sencilla llenado el campo y no se obtienen resultados. | El sistema no encuentra ningún documento que contenga los parámetros de búsqueda indicados. | V (Etiopia) | El sistema busca todas las coincidencias con la palabra y muestra el mensaje "La búsqueda no arrojó ningún resultado." | 1-Hacer clic en el Botón Búsqueda Sencilla 2- El usuario introduce el dato y presiona el botón Buscar. 3- El sistema muestra el mensaje "La búsqueda no arrojó ningún resultado." |
| EC 1.3 Realizar Búsqueda Sencilla dejando el campo vacío. | No se realiza la búsqueda | I (vacío) | El sistema no activa el botón Buscar. | 1-Hacer clic en el Botón Búsqueda Sencilla 2- El usuario deja el campo de búsqueda vacío. 3- No se activa el botón Buscar. |

Tabla 8: Diseño de casos de prueba para el CUS Realizar búsqueda sencilla

3.3.3. Pruebas de carga y estrés

Estas pruebas se diseñan para enfrentar a los sistemas a situaciones anormales, es decir, ejecutar el sistema en forma que demande recursos en cantidad, frecuencia o volúmenes anormales. Igualmente busca validar el correcto funcionamiento del sistema bajo las condiciones de carga normales para la operación, para concluir sobre variables como: el tiempo de respuesta, carga de procesamiento, trabajo por unidad de tiempo y utilización de recursos.

3.3.3.1. Resultados pruebas de carga y estrés

Las pruebas de rendimiento fueron realizadas en la herramienta JMeter propuesta por la Universidad de Ciencias Informáticas para pruebas de carga y estrés. El entorno en que fueron realizadas cumple con las siguientes características: 1 PC Cliente de 4 GB de memoria RAM y Sistema Nova 4.0, y servidor web y



de datos con Sistema Operativo Windows 8 de 4 GB de memoria RAM. En la siguiente tabla se muestran algunos de los resultados realizados a las principales funcionalidades del portal (Ver Tabla 9).

| Funcionalidad | URL | Respuesta/ Número de usuarios (segundos) | | Media Rendimiento (segundos) |
|--------------------------------|---|--|--------|------------------------------|
| | | 50 | 100 | |
| Realizar búsqueda sencilla | Dexcriba/index.php/busqueda/busqueda_sencilla | 0.1806 | 1.0226 | 0.6016 |
| Realizar búsqueda avanzada | Dexcriba/index.php/busqueda/busqueda_avanzada | 0.3107 | 1.4304 | 0.87055 |
| Realizar búsqueda exploratoria | Dexcriba/index.php/busqueda/busqueda_exploratoria | 0.5440 | 0.5440 | 0.5440 |

Tabla 9: Resultados pruebas de carga y stress

3.3.4. Resultado de las pruebas realizadas.

Con la realización de estas pruebas fue posible detectar algunas no conformidades, estas fueron eliminadas mediante el transcurso de cada una de las iteraciones definidas. Las mismas permitieron garantizar que el módulo cumpliera con las especificaciones que se trazaron de este modo dándole cumplimiento a cada uno de los requisitos definidos en el transcurso de la investigación. A continuación se representan las no conformidades que fueron detectadas y a su vez la solución a cada una de ellas en forma de iteración:



| No de la iteración | Fecha | Casos de prueba | No conformidades | | | Estado | | |
|--------------------|------------|--------------------------------|------------------|----|---|--------|----|----|
| | | | S | NS | R | RA | PD | NP |
| 1 | 10-05-2013 | Realizar búsqueda sencilla | - | 4 | - | 4 | - | - |
| 2 | 28-05-2013 | | - | - | - | - | - | - |
| 3 | 05-06-2013 | | | | | | | |
| 1 | 10-05-2013 | Realizar búsqueda avanzada | 3 | 4 | - | 7 | - | - |
| 2 | 28-05-2013 | | 1 | - | 1 | 2 | - | - |
| 3 | 05-06-2013 | | - | - | - | - | - | - |
| 1 | 10-05-2013 | Realizar búsqueda exploratoria | 1 | - | - | 1 | - | - |
| 2 | 28-05-2013 | | - | - | - | - | - | - |
| 3 | 05-06-2013 | | - | - | - | - | - | - |
| 1 | 10-05-2013 | Exportar resultados | 1 | - | - | 1 | - | - |
| 2 | 28-05-2013 | | - | - | - | - | - | - |
| 3 | 05-06-2013 | | - | - | - | - | - | - |
| 1 | 10-05-2013 | Importar resultados | - | - | - | - | - | - |
| 2 | 28-05-2013 | | - | - | - | - | - | - |
| 3 | 05-06-2013 | | - | - | - | - | - | - |
| 1 | 10-05-2013 | Ordenar resultados | 1 | - | 1 | 2 | - | - |
| 2 | 28-05-2013 | | - | - | - | - | - | - |
| 3 | 05-06-2013 | | - | - | - | - | - | - |

No conformidades: S – Significativa NS – No significativa R- Recomendación
Estado: RA – Resuelta PD – Pendiente NP - No procede

Tabla 10: Resultados de las pruebas realizadas

En lo anterior se puede observar que las pruebas realizadas se llevaron a cabo en tres iteraciones, donde se detectaron 17 no conformidades de las cuales 8 fueron bajas (no significativas), 7 altas (significativa) y 2 recomendaciones, quedando todas resueltas para la tercera iteración (ver Tabla 11), por lo tanto, se puede concluir que las pruebas en el módulo fueron realizadas satisfactoriamente quedando el sistema listo para su funcionamiento.

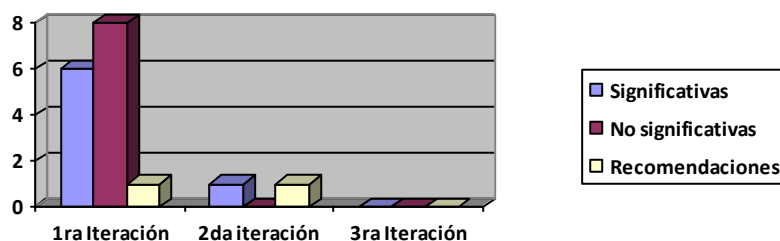


Tabla 11: Comparación entre los resultados obtenidos por iteración



3.4. Conclusiones parciales

Al finalizar el desarrollo de este capítulo se obtuvo un módulo completamente funcional. En este punto se encuentran implementados los elementos del diseño satisfactoriamente con los requerimientos definidos. Se generaron nuevos artefactos como los diagramas de componentes y el modelo de despliegue. Finalmente, se logró identificar las principales fortalezas a partir del análisis de los resultados arrojados una vez realizadas las pruebas al *software*.



Conclusiones generales

- El análisis de los mecanismos de recuperación en sistemas de gestión de documentos históricos tanto nacionales como internacionales evidenció la necesidad de implementar una solución a la medida para el SGDH Dexcriba, porque las soluciones encontradas no satisfacen por sí solas las necesidades del presente trabajo de diploma. Además, permitió identificar nuevos requisitos funcionales.
- El módulo para la recuperación de información implementado presentó un correcto funcionamiento de acuerdo con los resultados obtenidos a partir de las pruebas realizadas.
- La implementación del módulo Recuperación de información posibilitó un mayor acceso y difusión de las colecciones documentales almacenadas en el GDH Dexcriba.



Recomendaciones

Teniendo en cuenta las experiencias obtenidas con el desarrollo de la investigación se recomienda:

- Internacionalizar el módulo.
- Implementar el protocolo OAI-PMH³² para poner a disposición de otros sistemas el fondo documental contenido en Dexcriba y permitir que este pueda acceder a los metadatos descritos en sistemas homólogos.

³² El protocolo OAI-PMH (*Open Archives Initiative-Protocol Metadata Harvesting*), es una herramienta de interoperabilidad independiente de la aplicación que permite realizar el intercambio de información para que desde puntos (proveedores de servicio), se puedan hacer búsquedas que abarquen la información recopilada en distintos repositorios asociados (proveedores de datos)



Referencias Bibliográficas

Alonso Martínez, H. et al. *Desarrollo del Módulo de Reportes del Sistema Integrado de Gestión Bibliotecaria Koha para la Biblioteca Nacional de Cuba José Martí.* Universidad de las Ciencias Informáticas, 2010.

Apache.org. *The Apache Software Foundation.* [En línea]. 2013. [Citado el: 12 de enero de 2013] Disponible en: <http://www.apache.org>.

Archives, University of Illinois. *Archon. The Simple Archival Information System.* [En línea]. 2011. [Citado el: 25 de marzo de 2013]. Disponible en: <http://www.archon.org/>.

Archivo, 3000. [En línea]. 2013. [Citado el: 22 de marzo de 2013.] Disponible en: <http://www.archivo3000.com/>.

Baeza Yates, R., et al. *Modern information retrieval.* New York : ACM Press, 1999, 513 p. ISBN 0-201-39829-X.

Bahit, E. *El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC.* 2011. 66 p.

Barroso Horta, Lic. Lourdes. et al. *Archivo Histórico Provincial. Los Archivos Históricos como fuente de conocimiento y cultura para los estudiantes universitarios.* [En línea]. 2010. [Citado el: 05 de Octubre de 2011] Disponible en: <http://archivohistorico.villaclara.cu/estudios/los-archivos-historicos-y-las-universidades>.

Blair, D.C. *Language and representation in information retrieval.* Amsterdam: Elsevier Science Publishers, 1990.

Brito Acuña, K. *Selección de metodologías de desarrollo para aplicaciones Web en la facultad de informática de la Universidad de Cienfuegos.* Facultad de Informática, Carrera de Ingeniería Informática Universidad de Cienfuegos “Carlos Rafael Rodríguez” 2009.

Chowdhury, G. G. *Introduction to modern information retrieval.* London: Library Association, 1999.

Codelgniter. [En línea]. 2013. [Citado el: 23 de Octubre de 2012]. Disponible en:



http://codeigniter.com/user_guide/.

Consejo Internacional de Archivos. *Normal Internacional General de Descripción Archivística ISAD (G)*. Madrid, 2000.

Croft, W. B. *Approaches to intelligent information retrieval. Information Processing & Management*. 1987. p. 249-254.

Cruz Mundet, J. R. *Administración de documentos y archivos. Textos fundamentales*. Madrid: 2011. 468 p. ISBN 978-84-615-5150-7.

Date, C. J. *Introducción a los Sistemas de bases de datos*. México: Pearson Education, 2001.

Diccionario de la Real Academia (RAE). [En línea]. 2013. [Citado el: 17 de Diciembre de 2012]. Disponible en: <http://www.rae.es>.

Fernández Villaverde, Raimundo. *Albalá*. [En línea]. 2013. [Citado el: 22 de marzo de 2013]. Disponible en: <http://www.baratz.es/Soluciones/Archivos/Albalá>.

Fuster Ruiz, Francisco. *Archivo, Documento de archivo. Necesidad de clarificar los conceptos*. [En línea]. 2006. [Citado el: 11 de noviembre de 2012.]. Disponible en: <http://www.slideshare.net/veronicaguz22/archivo-documento-de-archivo-necesidad-de-clarificar-los-conceptos>.

Gamma, Erich. *Patrones GoF*. [En línea]. 2010. [Citado el: 24 Mayo 2013]. Disponible en: <http://geektheplanet.net/5462/patrones-gof.xhtml>.

García, Francisco J., et al. *Ingeniería de Software. Diseño orientado a objetos*. [En línea]. 2008. Disponible en: <http://ocw.usal.es/enseanzas-tecnicas/ingenieria-del-software/contenidos/Tema6-DOO-1pp.pdf>.

Glosario A.L.A. *Bibliotecología y Ciencias de la Información*. Madrid: Díaz de Santos, 1983.

Grossman, D.A., et al. *Information retrieval: algorithms and heuristics*. Boston: Kluwer Academia Publishers, 1998.

Gomez Diaz, R., et al. *Normalización y requisitos funcionales de la descripción archivística: una propuesta metodológica*. Scire: Representación y organización del conocimiento. [En línea]. 2005. [Citado el: 15 de



marzo de 2013]. Disponible en: <http://eprints.rclis.org/handle/10760/13665>. ISSN 1135-3761.

Heredia Herrera, Antonia. *Archivística General Teoría y Práctica*. Quinta edición. Diputación Provincial de Sevilla, 1991. ISBN 84 - 7798 - 056 - X.

ICA Atom. [En línea]. 2013. [Citado el: 30 de febrero de 2013]. Disponible en: <http://ica-atom.org/>.

IEEE. *IEEE: Guide for Developing System Requirements Specification*. 1998. ISBN 1-55937-716-X. [Consultado el: 12 de marzo 2012].

Issi Camy, Lázaro. *JavaScript*. Madrid : Anaya Multimedia, 2002. ISBN: 84-415-1384-8.

Jacobson, I, et al. *El proceso unificado de desarrollo de software*. Madrid: 2000. vol. 2012, ISBN 84-7829-036-2.

Jquery. *User interface*. [En línea]. 2012. [Citado el: 29 de marzo de 2013]. Disponible en: <http://blog.jqueryui.com/2012/11/jquery-ui-1-9-2/>.

Korfhage, R.R. *Information Retrieval and Storage*. New York: Wiley Computer Publisher, 1997.

Longley, D, et al. *Dictionary of IT*. London and Basingstoke: The MacMillan Press, 1989.

Marquina, E., et al. *Guía de Patrones, Prácticas y Arquitectura.NET*. [En línea]. La Habana, 2008. [Consultado el: 17 de abril de 2013]. Disponible en: <http://sunshine.prod.uci.cu/gridfs/sunshine/books/PPArquitecturaNET.pdf>.

Martínez Méndez, Franciso Javier. *Recuperación de información: modelos, sistemas y evaluación*. [En línea]. 2004. [Citado el: 3 de abril de 2013]. Disponible en: <http://digitum.um.es>. ISBN: 84-932537-7-4.

Mato García, R. M. *Diseño de Bases de datos*. [En línea]. 1999. [Consultado el: 11 de marzo de 2013]. Disponible en: Http://eva.uci.cu/mod/resource/view.php?id=11576&subdir=/Bibliografia_Basica.

Mena Mugica, Mayra M, et al. *Evolución de la ciencia Archivística*. [En línea]. 2009. [Citado el: 13 de noviembre de 2012.] Disponible en: http://bvs.sld.cu/revistas/aci/vol20_1_09/aci04709.htm. ISSN 1024-9435.

Mena Mugica, Mayra M. *Gestión documental y organización de los archivos*. La Habana: 2005. 96 p. ISBN 959-258-950-X.



Menéndez Barzanallana, Asensio R. *Informática Aplicada a la Gestión Pública*. Facultad Derecho UMU. [En línea]. 2011. [Citado el: 24 de abril de 2013]. Disponible en: <http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html>.

Ministerio de cultura. *Archivística*. [En línea]. 1995. [Citado el: 03 de marzo 2013]. Disponible en: <http://www.mcu.es/archivos/MC/DTA/Diccionario.html#archiv%C3%ADstica>. ISBN 84-8181-066-5.

Mora, Raúl, et al. *Desarrollo de herramienta software para el apoyo en la comprensión de estructuras básicas de programación*. [En línea]. 2010. [Citado: el 8 de marzo de 2013]. Disponible en: http://cecip.upaep.mx/titulacion_sep/docs/LI2EJ1.pdf.

Murillo Alfaro, Félix. *Instituto Nacional de Estadística e Informática. Herramientas Case*. [En línea]. Perú: 1999. [Citado el: 29 de marzo de 2013]. Disponible en: <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>.

Murphey, Rebecca. *JQuery fundamentals*. [En línea] 2010. [Citado el: 23 de marzo de 2013]. Disponible en: <http://www.etnassoft.com/biblioteca/jquery-fundamentals>.

NetBeans. *Qué es NetBeans?*. [En línea]. 2011. [Citado el: 21 de mayo de 2013.] Disponible en: http://netbeans.org/index_es.html.

Pares. [En línea]. 2013. [Citado el: 22 de abril de 2013]. Disponible en: <http://www.pares.es/>.

Pérez Carballo, J, et al. *Natural language information retrieval: progress report*. Information Processing and Management 36, 2000. p. 155-178.

Pérez, Javier Eguíluz. *Introducción a AJAX*. [En línea]. 2010. [Citado el: 30 de abril de 2013]. Disponible en: <http://www.librosweb.es/css/capitulo1.html> I REFERENCIAS.

PHP. *General Information*. [En línea] 2011. [Citado el: 22 de abril de 2013]. Disponible en: <http://www.php.net/manual/en/faq.general.php#faq.general.what>.

PostgreSQL. *Introducción a PostgreSQL*. [En línea]. Perú, 2007. [Citado el: 3 de febrero de 2013.] http://postgresql.org.pe/articles/introduccion_a_postgresql.pdf.

Pressman, Roger. *Ingeniería del Software. Un enfoque práctico*. Quinta edición. La Habana : McGraw-Hill, 2002.



- Pressman, Roger.** *Ingeniería de Software. Un enfoque práctico.* Madrid y Carachelejo (España) : s.n., 2001.
- Reyes, Cecilia.** *Guía de Estandares de Codificación.* [En línea]. 2003. [Citado el: 2 de mayo de 2013]. Disponible en: <http://www.inf.utfsm.cl>.
- Rijsbergen, C.J.** *Information Retrieval.* Glasgow, University. [En línea]. 1999. [Citado el: 21 de enero de 2013]. Disponible en: <http://www.dcs.gla.ac.uk/~iain/keith/>.
- Riveros, Héctor, et al.** *El método científico aplicado a las ciencias experimentales.* México : Trillas, 2000. ISBN 968-24-3893-4.
- Salton, G., et al.** *Introduction to Modern Information Retrieval.* New York: Mc Graw-Hill Computer Series, 1983.
- Santos, Prof. MSc. Rafael Oliva.** *Introducción a los Sistemas de recuperación de Información.* Habana: Departamento de Ciencia de la Computación UH, 2009.
- Tramullas Sáez, J.** *Introducción a la Documática.* [En línea]. Zaragoza: Kronos, 1997. [Citado el: 18 de febrero de 2013]. Disponible en: <http://www.tramullas.com/nautica/documatica/3-1.html>.
- UML.** *Lenguaje de Modelo UML.* [En línea]. 2011. [Citado el: 3 de Noviembre de 2012]. Disponible en: <http://www.docirs.cl/uml.htm>.
- Unidad,1.** *Los Lenguajes de Marcas.* [En línea]. 2009. [Citado el: 30 abril 2013]. Disponible en: <http://www.scribd.com/doc/37990287/UNIDAD-1-Los-Lenguajes-de-Marcas>.
- World Wide Web (w3).** *Tutorial de AJAX.* [En línea]. 2010. [Citado el: 30 de marzo de 2013.] Disponible en: <http://www.w3schools.com/ajax/default.asp>.
- World Wide Web Consortium (w3c).** *HTML & CSS.* [En línea]. 2012. [Citado el: 26 de febrero de 2013]. Disponible en: <http://www.w3.org/standards/webdesign/htmlcss>.
- World Wide Web (w3).** *Extensible Markup Language (XML).* [En línea]. 2010. [Citado el: 22 de marzo de 2013.]. Disponible en: <http://www.w3.org/XML/>.



Bibliografía

Pressman, Roger S. *Ingeniería de Software, un enfoque práctico*. Quinta edición: S.I: McGraw-Hill Companies, 2002. ISBN 8448132149.

Luna Hernández, D. F. *La importancia del aseguramiento de la calidad en los sistemas de información*. Revista Ingeniería primero, octubre 2010 2010, nº 19, p. 45-48. [En línea]. [Citado el: 30 de Octubre de 2011]. Disponible en: http://www.tec.url.edu.gt/boletin/URL_19_SIS09_GESTION.pdf. ISSN 2076-3166.

Fernandez Valderrama, Luis David. *Gestión Documental*. 2007. [En línea]. [Citado el: 30 de Octubre de 2011.] Disponible en: [http://www.sociedadelainformacion.com/12/Gestion Documental.pdf](http://www.sociedadelainformacion.com/12/Gestion%20Documental.pdf).

Peña, Yuricel. *Propuesta de proceso macro de Monitoreo y Control para Nivel 2 de CMMI*. 2008. Disponible en: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_1426_08.

Erasmus, Mundus. *Validacion de requisitos*. 2006. [En línea]. [Citado el: 30 de Octubre de 2011]. Disponible en :http://is.ls.fi.upm.es/docencia/masterTI/ARS/docs/Manual_M2C1U11.pdf.

Parra, José David. *Guía de Patrones, Prácticas y Arquitectura .NET*. 2008. [En línea]. [Citado el: 6 de febrero de 2012]. Disponible en: <http://sunshine.prod.uci.cu/gridfs/sunshine/books/PPArquitecturaNET.pdf>

Álvarez, Miguel Ángel. *Manual de CodeIgniter*. [Citado el: 12 de enero 2013]. Disponible en: <http://sunshine.prod.uci.cu/gridfs/sunshine/books/manual-codeigniter.pdf>



Anexos

Anexo 1. Modelo de la entrevista aplicada

| Entrevista | | | |
|---|--|----------------------------|--|
| Objetivo | Detectar las deficiencias y necesidades al trabajar con los mecanismos de búsqueda en sistemas para la gestión de documentos históricos. | | |
| Temática que se investiga | Creación de un módulo de recuperación de la información para el sistema de gestión de documentos históricos Describa. | | |
| Usted ha sido seleccionado por sus conocimientos sobre gestión documental y archivística, por sus años de experiencia en el área y los resultados alcanzados en su labor profesional; como experto a evaluar las necesidades prioritarias en cuanto al trabajo con sistemas de gestión de documentos históricos. Se le agradece de antemano su cooperación. | | | |
| Nombre y Apellidos | | | |
| Rol en que se desempeña | | Años de experiencia | |
| Preguntas | | | |
| <ol style="list-style-type: none"> 1. ¿Cuáles son las expectativas existentes para el nuevo módulo de recuperación de la información en el SGDH Describa, sobre su predecesor en ArchivenHis? 2. ¿Qué elementos considera usted que debería tener el módulo para la recuperación de información? 3. ¿Cree necesario que los criterios de búsqueda abarquen áreas no contempladas en la norma ISAD (G)? 4. ¿Cree necesario realizar salvadas de los resultados de las búsquedas? ¿Por qué? | | | |

Tabla 12: Formato de la entrevista aplicada

Anexo 2. Descripciones de los CUS

| Caso de Uso: | CU1 Realizar Búsqueda Sencilla | | |
|---------------------------------------|--|---|--|
| Actores: | Usuario (Inicia). | | |
| Resumen: | El caso de uso inicia cuando el actor procede a realizar una búsqueda general sobre los documentos contenido en el sistema y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso. | | |
| Precondiciones: | | | |
| Referencias | RF1,RF4,RF5 | | |
| Prioridad | Crítica | | |
| Flujo Normal de Eventos | | | |
| Acción del Actor | | Respuesta del Negocio | |
| 1-Introduce el parámetro de búsqueda. | | 1.1 Busca en el sistema las descripciones cuyos datos concuerden con el introducido y muestra todos los posibles resultados debajo de los parámetros. | |
| Flujos Alternos | | | |



| | |
|--|--|
| Línea 1.1: El sistema no encuentra ninguna descripción cuyos datos coincidan con el parámetro introducido. | 1.1 El sistema informa que la búsqueda no arrojó ningún resultado. |
| Línea 1.2: No introduce ningún en el campo de búsqueda | 1.2- No se activa el botón Buscar. |
| Línea 1.3: No presiona el botón Buscar. | 1.3- Queda cancelada la opción de búsqueda. |
| Pos condiciones | Se muestra una lista de documentos presentes en el sistema cuyas descripciones contienen algún campo que coincide con el criterio de búsqueda proporcionado. |

Tabla 13: Descripción del CUS Realizar búsqueda avanzada

| | |
|---|--|
| Caso de Uso: | CU1 Realizar Búsqueda Exploratoria |
| Actores: | Usuario (Inicia). |
| Resumen: | El caso de uso inicia cuando el actor procede a realizar una búsqueda sobre los documentos contenidos en el sistema, ubicándolos directamente por su localización física, así finaliza el caso de uso. |
| Precondiciones: | |
| Referencias | RF3,RF4,RF5. |
| Prioridad | Crítica. |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Negocio |
| 1-Selecciona el nivel más alto dentro del árbol de las ubicaciones físicas. | 1.1 El sistema muestra los subniveles correspondientes al seleccionado, donde el usuario podrá acceder al siguiente nivel. |
| 2- Selecciona el documento que buscaba dentro del árbol. | El sistema muestra la descripción asociada a ese documentos y las opciones pertinentes. |
| Pos condiciones | Se muestra la descripción detallada del documento localizado directamente por su ubicación física. |

Tabla 14: Descripción del CUS Realizar búsqueda exploratoria

| | |
|---|--|
| Caso de Uso: | CU4 Ordenar resultados de las búsquedas. |
| Actores: | Usuario (Inicia). |
| Resumen: | El caso de uso es una inclusión del CU1 o CU2, el sistema visualiza los resultados obtenidos sobre él o los parámetros y el actor puede ordenarlos en dependencia de los seis campos establecidos por la norma ISAD (G). |
| Precondiciones: | Se debe haber realizado alguna búsqueda o importado un paquete de búsquedas previamente creado. |
| Referencias | RF6,RF7,RF8, RF9, RF10. |
| Prioridad | Secundario |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Negocio |
| 1- Selecciona la columna de acuerdo al criterio para ordenar. | 1.1-Ordena acorde a la columna seleccionada los resultados obtenidos |



| Flujos Alternos | |
|--|---------------------------------------|
| Línea 1: El usuario no realiza el proceso de ordenamiento. | |
| Poscondiciones | Resultados de las búsquedas ordenado. |

Tabla 15: Descripción del CUS Ordenar resultados de las búsquedas

| Caso de Uso: | CU5 Exportar resultados de las búsquedas |
|---|---|
| Actores: | Usuario (Inicia). |
| Resumen: | El caso de uso es una inclusión del CU1 o CU2, y es el encargado de guardar en un fichero XML los datos almacenados como resultado del proceso de búsqueda. |
| Precondiciones: | Se debe haber realizado una búsqueda previamente |
| Referencias | RF11. |
| Prioridad | Apoyo |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Negocio |
| 1- Selecciona la opción Exportar en la interfaz de búsqueda. | 1.1 Muestra la ventana Abriendo búsqueda. |
| 2- Selecciona la dirección en la cual guardar los datos y proporciona el nombre del archivo donde se guardarán los datos. | |
| 3- Pulsar Aceptar para guardar los datos. | 4-Guarda los datos satisfactoriamente |
| Poscondiciones | Como resultado se crea una salva con los resultados de la búsqueda. |

Tabla 16: Descripción del CUS Exportar resultados de las búsquedas

| Caso de Uso: | CU6 Importar resultados de las búsquedas |
|--|--|
| Actores: | Usuario (Inicia) |
| Resumen: | El caso de uso inicia cuando el actor procede a importar los resultados de una búsqueda previamente realizada, el sistema carga el fichero que contiene dicha información y la muestra en la página de búsqueda. Una vez ejecutada esta acción por parte del sistema, finaliza el caso de uso. |
| Precondiciones: | Debe existir un fichero de búsqueda previamente creado. |
| Referencias | RF12. |
| Prioridad | Apoyo |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Negocio |
| 1- Selecciona la opción Importar en la interfaz de búsqueda. | 1.1 Muestra la ventana Carga de archivos. |
| 2- Selecciona el archivo de datos a importar. | |
| 3- Pulsar Abrir para cargar los datos. | |
| | 4- Se cargan en la aplicación los datos importados. |
| Flujos Alternos | |



| | |
|--|---|
| Línea 2: La dirección especificada no tiene ficheros que importar. | |
| Poscondiciones | Se importa el archivo de datos y se muestran los resultados en la página de búsqueda. |

Tabla 17: Descripción del CUS Importar resultados de las búsquedas



Anexo 3. Diagramas de clases del análisis

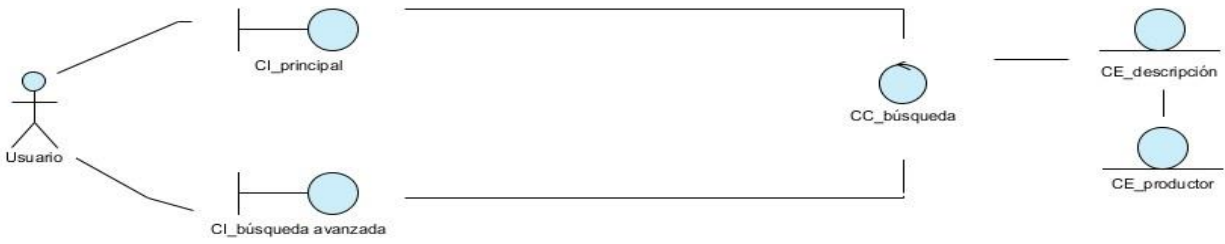


Ilustración 12: Diagrama de clases del análisis del CUS Realizar búsqueda avanzada

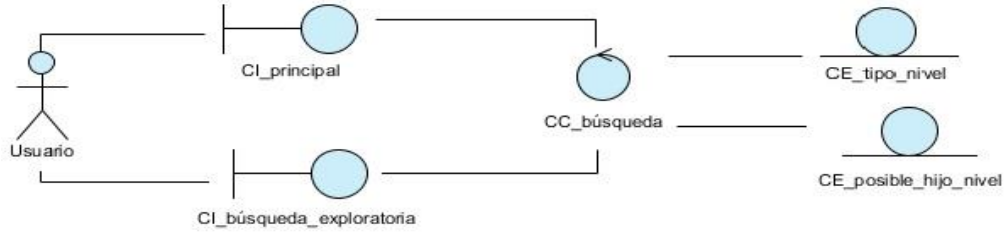


Ilustración 13: Diagrama de clases del análisis del CUS Realizar búsqueda exploratoria

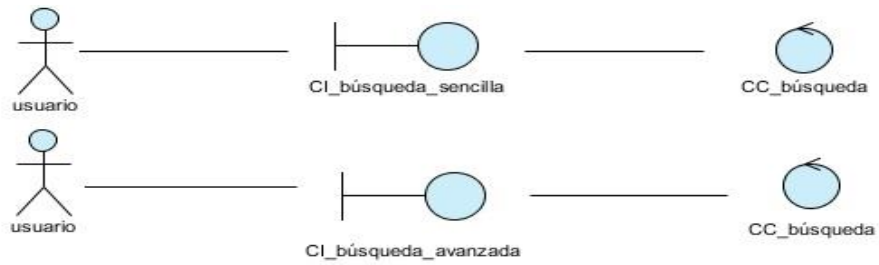


Ilustración 14: Diagrama de clases del análisis del CUS Exportar resultados de la búsqueda

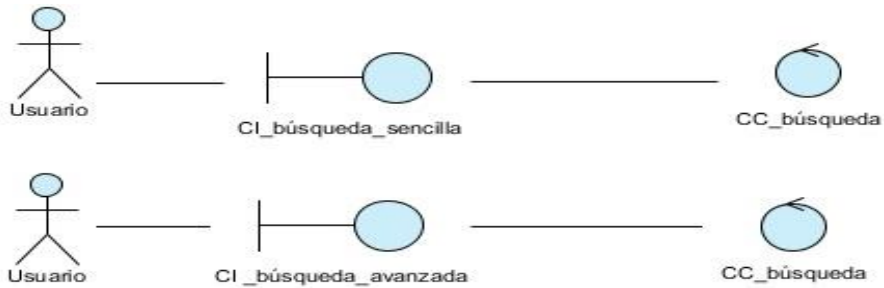


Ilustración 15: Diagrama de clases del análisis del CUS Importar resultados de la búsqueda



Anexo 4. Diagramas de interacción

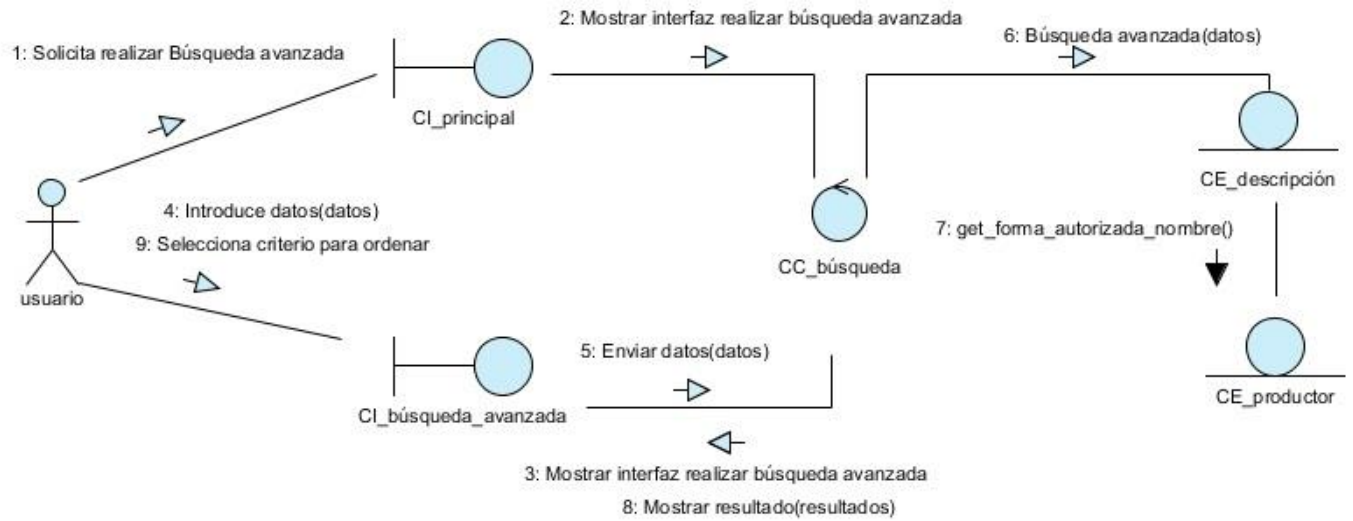


Ilustración 16: Diagrama de interacción del CUS Realizar búsqueda avanzada

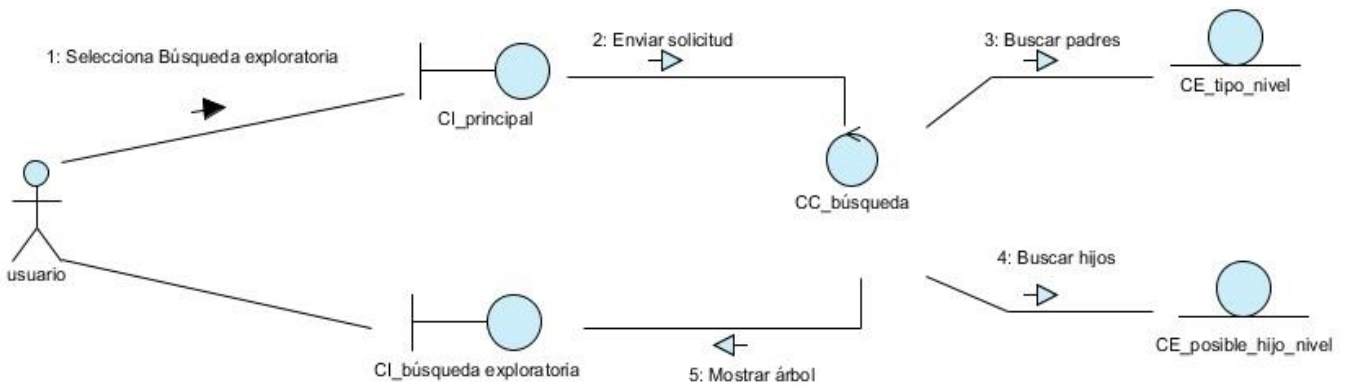


Ilustración 17: Diagrama de interacción del CUS Realizar búsqueda exploratoria

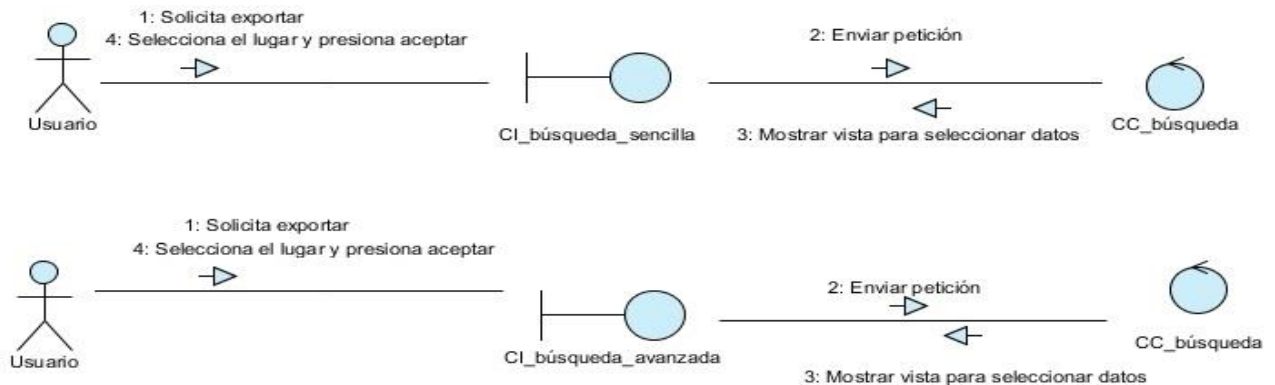


Ilustración 18: Diagrama de interacción del CUS Exportar resultados de la búsqueda



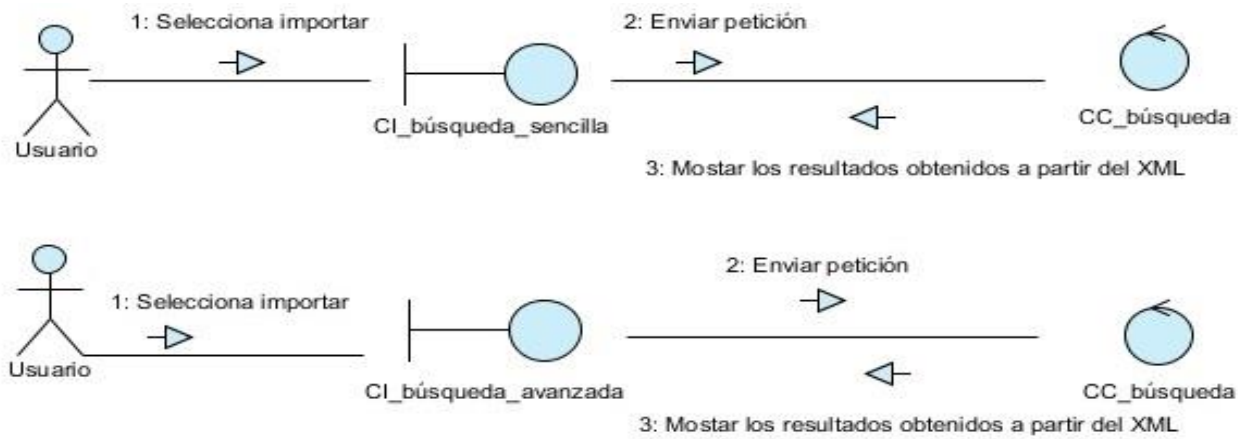


Ilustración 19: Diagrama de interacción del CUS Importar resultados de la búsqueda

Anexo 5. Modelo de datos del esquema Descripción

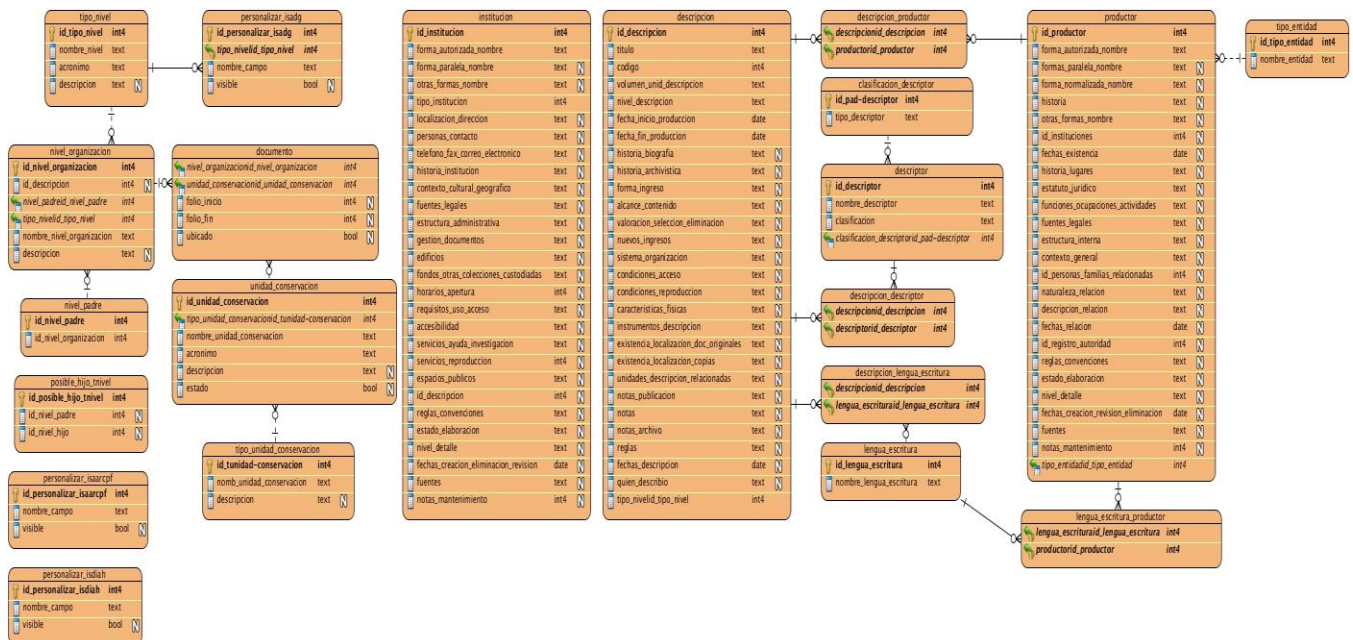


Ilustración 20: Modelo de datos del esquema Descripción



Descripción de las entidades utilizadas del modelo descripción

| | | |
|--------------------------|--|---|
| Tabla | descripción | |
| Descripción | Almacena los datos de que se utilizan para describir el nivel de organización. | |
| Atributos | Tipo | Descripción |
| id_descripcion | integer(10) | Identifica las tuplas en la tabla, incremental generado por el gestor de base de datos. |
| titulo | text | Representa el título de la descripción. |
| codigo | integer(10) | Representa el código que recibe la descripción. |
| volumen_unid_descripcion | text | Indica el volumen de descripción en el que se encuentra el nivel de organización. |
| nivel_descripcion | | Indica el nivel de descripción en que se encuentra el nivel de organización. |
| fecha_inicio_produccion | date | Representa la fecha inicial de producción de la descripción. |
| fecha_fin_produccion | date | Representa la fecha final de producción de la descripción. |

Tabla 18: Descripción de la entidad descripción correspondiente al modelo Descripción

| | | |
|---------------------------|---|--|
| Tabla | descripcion_productor | |
| Descripción | Almacena la relación existente entre la descripción y el productor. | |
| Atributos | Tipo | Descripción |
| descripcionid_descripcion | integer(10) | Representa el identificador de la descripción. |
| productorid_productor | integer(10) | Identifica el productor. |

Tabla 19: Descripción de la entidad descripcion_productor correspondiente al modelo Descripción

| | | |
|-----------------------------|--|---|
| Tabla | productor | |
| Descripción | Almacena los datos de que se utilizan para describir el productor. | |
| Atributos | Tipo | Descripción |
| id_productor | integer(10) | Identifica las tuplas en la tabla, incremental generado por el gestor de base de datos. |
| forma_autorizada_nombre | text | Identifica de manera unívoca al productor. |
| tipo_entidadid_tipo_entidad | integer(10) | Representa el identificador |

Tabla 20: Descripción de la entidad productor correspondiente al modelo Descripción

| | |
|--------------|--------------|
| Tabla | tipo_entidad |
|--------------|--------------|



| | | |
|--------------------|---|---|
| Descripción | Almacena los datos del tipo de entidad. | |
| Atributos | Tipo | Descripción |
| id_tipo_entidad | integer(10) | Identifica las tuplas en la tabla, incremental generado por el gestor de base de datos. |
| nombre_entidad | text | Representa el nombre que recibe la entidad |

Tabla 21: Descripción de la entidad tipo_entidad correspondiente al modelo Descripción

| | | |
|--------------------|---|---|
| Tabla | tipo_nivel | |
| Descripción | Clasificación de los niveles de organización. | |
| Atributos | Tipo | Descripción |
| id_tipo_nivel | integer(10) | Identifica las tuplas en la tabla, valor incremental generado por el gestor de base de datos. |
| nombre_nivel | text | Representa el nombre del nivel. |
| acronimo | text | Identificador asociado al nombre del tipo de nivel. |
| Descripcion | text | Representa la descripción del tipo de nivel. |

Tabla 22: Descripción de la entidad tipo_nivel correspondiente al modelo Descripción

| | | |
|------------------------|--|---|
| Tabla | posible_hijo_tnivel | |
| Descripción | Representa la posible relación entre un nivel de organización padre (raíz) y un nivel de organización hijo (nodo). | |
| Atributos | Tipo | Descripción |
| id_posible_hijo_tnivel | integer(10) | Identifica las tuplas en la tabla, incremental generado por el gestor de base de datos. |
| id_nivel_padre | integer(10) | Representa el identificador del nivel de organización padre (raíz). |
| id_nivel_hijo | integer(10) | Representa el identificador del nivel de organización hijo (nodo). |

Tabla 23: Descripción de la entidad posible_hijo_nivel correspondiente al modelo Descripción

Anexo 6. Método enumerado para realizar la técnica del camino básico

```

Public function busqueda_sencilla()
{
1  $data['busqueda_sencilla'] = $this->lang->line('busqueda_sencilla');
  $data['enviar'] = $this->lang->line('enviar');
  $data['exportar'] = $this->lang->line('exportar');
  $data['importar'] = $this->lang->line('importar');
2  if ($this->input->post('submit_btn')) {

```




```

3     $data['resultados'] = $this->mdl->Busqueda_Sencilla($this->input->post('cadena_txt')); }
4     if ($this->input->post('imp') and isset($_FILES['fileInput'])) {
5         $file = fopen($_FILES['fileInput']['tmp_name'], 'r');
           $xml = fread($file, filesize($_FILES['fileInput']['tmp_name']));
           fclose($file);
           $data['importado'] = $this->mdl->Importar_Busqueda($xml); }
6     $this->template->add_js('ui/jquery.ui.position.min');
           $this->template->add_js('ui/jquery.ui.tooltip.min');
           $this->template->add_js('ui/jquery.ui.mouse.min');
           $this->template->add_js('ui/jquery.ui.draggable.min');
           $this->template->add_js('ui/jquery.ui.resizable.min');
           $this->template->add_js('ui/jquery.ui.dialog.min');
           $this->template->add_js('modulos/busqueda/dataTables');
           $this->template->add_js('modulos/busqueda/ColReorder');
           $this->template->add_js('modulos/busqueda/ColVis');
           $this->template->add_js('modulos/busqueda/vista_tabla');
           $this->template->add_js('modulos/busqueda/busqueda_simple');
           $this->template->add_js('modulos/busqueda/AjaxUpload');
           $this->template->add_css('modulos/busqueda/VisTabla');
           $this->template->set_content('busqueda_sencilla', $data);
           $this->template->build();
}

```

Tabla 24: Enumeración del método búsqueda sencilla

Anexo 7. Diseño de casos de prueba de caja negra

CUS Realizar búsqueda avanzada

| VX | Nombre del campo | Clasificación | Valor nulo | Descripción |
|----|-----------------------------|----------------|------------|--|
| V1 | Título | Campo de texto | Sí | Cadena de caracteres, título de la entidad que se desea buscar. |
| V2 | Código | Campo de texto | Sí | Cadena de caracteres, código que representa la entidad que se desea localizar. |
| V3 | Fecha descripción (inicial) | Calendario | Sí | Campo que tiene definido por defecto un calendario, donde el usuario localizará la fecha de descripción inicial del documento que desea encontrar. |
| V4 | Fecha descripción (final) | Calendario | Sí | Campo que tiene definido por defecto un calendario, donde el usuario localizará la fecha de descripción final del documento que desea encontrar. |
| V5 | Productor | Combobox | Sí | Campo que tiene definido por defecto los productores que al menos tienen asociado un documento. |
| V6 | Volumen unidad descripción | Combobox | Sí | Campo que tiene definido por defecto los volúmenes de las unidades de conservación existentes en el sistema. |
| V7 | Nivel de descripción | Combobox | Sí | Campo que tiene definido por defecto los niveles de descripción de las unidades de conservación existentes en el sistema. |

Tabla 25: Valores de entrada de datos para el caso de prueba del CUS Realizar búsqueda avanzada



| Escenario | Descripción | V1 | V2 | V3 | V4 | V5 | V6 | V7 | Respuesta del sistema | Flujo central |
|---|--|----------------------|---------------------|--------------------------------|--------------------------------|----------------------|----------------------|------------------------|---|--|
| EC 1.1 Realizar Búsqueda avanzada obteniendo resultados. | Busca en el sistema las descripciones de los documentos cuyos datos concuerden con los introducidos y muestra todos los posibles documentos. | V (Marti) | V (1233) | V (2013-01-05) | V (2016-01-05) | V (Fondo) | V (Nivel) | V (Carlos) | Muestra una lista con todos los documentos que coinciden con todos los criterios especificados. | 1-Hacer clic en Búsqueda avanzada 2- El usuario llena los campos y presiona el botón "Enviar". 3- El sistema muestra una lista con los resultados. |
| 1.2 Realizar Búsqueda avanzada dejando todos los campos vacíos | Busca en el sistema las descripciones de los documentos cuyos datos concuerden con los introducidos y muestra todos los posibles documentos. | V(vacío) | V(vacío) | V(vacío) | V(vacío) | V(vacío) | V(vacío) | V(vacío) | Muestra una lista con todos los documentos contenidos. | 1-Hacer clic en Búsqueda avanzada 2- El usuario deja todos los campos vacíos y presiona el botón "Enviar". 3- El sistema muestra una lista con todos los documentos. |
| EC 1.3 Realizar Búsqueda avanzada sin resultados | El sistema no encuentra ningún documento que contenga los parámetros de búsqueda indicados. | V(La isla) | V(vacío) | V(10/09/2000) | V(15/10/2011) | V(Todos) | V(Todos) | V(Todos) | El sistema muestra el mensaje "La búsqueda no arroja ningún resultado" | 1-Hacer clic en Búsqueda avanzada 2- El usuario llena los campos. 3-El sistema muestra el mensaje "La búsqueda no arroja ningún resultado" |
| EC 1.4 Realizar Búsqueda avanzada dejando campos vacíos. | Busca en el sistema las descripciones de los documentos cuyos datos concuerden | V(vacío) V(Marti) | V(1233) V(vacío) | V(2013-01-05) V(2013-01-05) | V(2016-01-05) V(2016-01-05) | V(Todos) V(Todos) | V(Todos) V(Todos) | V(Carlos) V(Carlos) | Muestra una lista con todos los documentos | 1-Hacer clic en Búsqueda avanzada 2- El usuario introduce los datos y presiona el botón "Enviar". |



| | | | | | | | | | |
|---|--|--|--|--|--|--|--|---|--|
| con los introducidos y muestra todos los posibles documentos debajo de los campos de búsqueda | | | | | | | | que coincida con los criterios especificados. | 3- El sistema muestra una lista con los resultados que coinciden con los parámetros de búsqueda. |
|---|--|--|--|--|--|--|--|---|--|

Tabla 26: Diseño de casos de prueba para el CUS Realizar búsqueda avanzada

CUS Exportar resultados de la búsqueda

| Escenario | Descripción | Respuesta del sistema | Flujo central |
|--------------------------|--|---|--|
| EC 1.1 Exportar búsqueda | Se guarda la búsqueda realizada para utilizarla en otro momento. | Guarda la búsqueda realizada en un XML. | 1- Selecciona la opción Exportar en la interfaz de los resultados de la búsqueda. 2- Selecciona la dirección y proporciona el nombre del archivo donde se guardarán los datos. 3- Pulsar Aceptar para guardar los datos. |

Tabla 27: Diseño de casos de prueba para el CUS Exportar resultados de la búsqueda

CUS Importar resultados de la búsqueda

| Escenario | Descripción | Respuesta del sistema | Flujo central |
|--|--|--|--|
| EC 1.1 Importar resultado de la búsqueda | El sistema carga la búsqueda guardada previamente. | El sistema carga la búsqueda guardada y muestra los resultados contenidos. | 1-El usuario selecciona la opción Importar la interfaz de búsqueda sencilla o avanzada. 2-Selecciona el archivo de datos a importa 3-Pulsar Aceptar. |

Tabla 28: Diseño de casos de prueba para el CUS Importar resultados de la búsqueda

CUS Realizar búsqueda exploratoria

| Escenario | Descripción | Respuesta del sistema | Flujo central |
|---------------------------------------|---|---|---|
| EC 1.1 Realizar Búsqueda exploratoria | Se despliega el árbol en dependencia de lo que el usuario valla seleccionando | Muestra los subniveles correspondientes al seleccionado, donde el usuario podrá seleccionar nuevamente el siguiente nivel hasta obtener los documentos contenidos en el último nivel y al pasar el cursor por encima se le muestra una descripción del mismo. | 1-El usuario selecciona la opción Búsqueda exploratoria. 2-El usuario selecciona el nivel más alto dentro del árbol de las ubicaciones físicas y va bajando en el árbol. 3-El usuario encuentra el documento deseado. |

Tabla 29: Diseño de casos de prueba para el CUS Realizar búsqueda exploratoria

CUS Ordenar resultados de la búsqueda



| Escenario | Descripción | Respuesta del sistema | Flujo central |
|--|---|---|---|
| EC 1.1 Ordenar resultado de la búsqueda | El sistema permite al usuario ordenar los resultados obtenidos a partir de una búsqueda sencilla o avanzada | El sistema permite al usuario definir cual de los criterios de búsqueda utilizados será el factor de ordenamiento, y permite hacerlo ascendentemente y viceversa. | 1-El usuario selecciona el nombre de la columna que contiene el criterio por el que desea ordenar en la interfaz de búsqueda sencilla o avanzada. |

Tabla 30: Diseño de casos de prueba para el CUS Ordenar resultados de la búsqueda

