

Universidad de las Ciencias Informáticas

Facultad 2



Título:

**Módulo de búsqueda y navegación de  
documentos de XABAL Repxos en la  
tecnología de Alfresco Community 5.2**

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

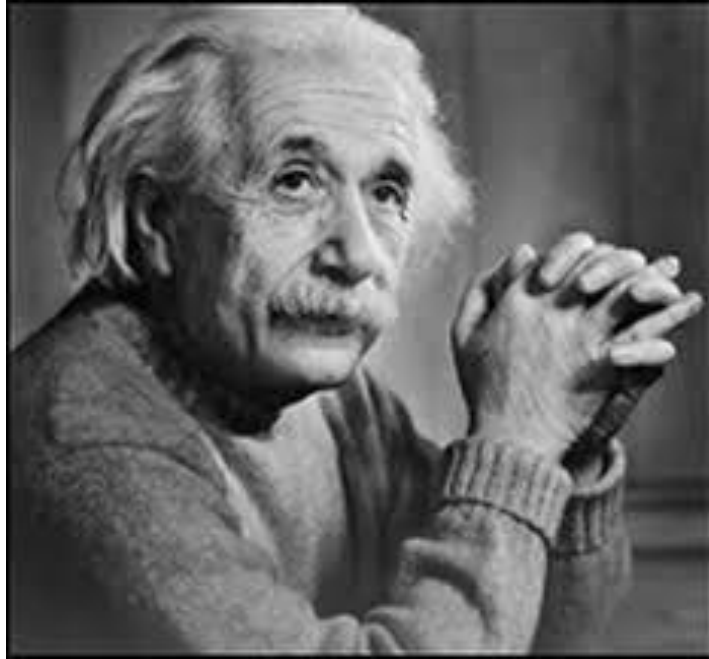
Autor:

Adriel Trujillo Pérez

Tutores:

Ing. Adrián Avilés Ávila

La Habana, junio 2019  
Año 61 de la Revolución.



*“El verdadero signo de la inteligencia no es el conocimiento, sino la imaginación”.*

*Albert Einstein*

## Declaración de autoría

Declaro ser autor del trabajo de diploma Módulo de búsqueda y navegación de documentos de XABAL Repxos en la tecnología de Alfresco Community 5.2. Reconozco a la Universidad de las Ciencias Informáticas (UCI), los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

Firma del Tutor

Ing. Adrián Avilés Ávila

\_\_\_\_\_

Firma del Autor

Adriel Trujillo Pérez

## Agradecimientos

*Agradezco a todos os que hicieron posible la realización de esta tesis, en especial a mi familia y amigos por darme ánimos y ayudarme a seguir adelante.*

## Dedicatoria

*Dedico esta tesis a mi familia y en especial a mi madre que ha sido siempre mi motor impulsor.*

## Resumen

XABAL Repxos 3.0 es un sistema para la implantación de repositorios digitales en instituciones científicas y académicas, desarrollado en el centro de Informatización de la Gestión Documental, perteneciente a la Universidad de las Ciencias Informáticas. Toda la información generada en los distintos tipos de eventos que se desarrollan en la universidad, es almacenada en XABAL Repxos; su integración con los procesos investigativos, permite utilizar la documentación como fuente de información, ofreciendo una vía para la divulgación del conocimiento y también una herramienta en el proceso docente-educativo.

La gran cantidad de información almacenada en XABAL Repxos hace necesario la existencia de un buscador para obtener los resultados deseados en el menor tiempo posible, este repositorio está montado sobre la tecnología que brinda Dspace, el cual incluye su propio buscador para satisfacer esta necesidad, como iniciativa se quiere cambiar toda la tecnología del proyecto a Alfresco que es una tecnología más reciente, es más sencilla, más potente, facilita la relación entre todos los componentes de los distintos productos y posee un potente visor de documentos para todos los formatos.

Debido a esta situación existente se decidió desarrollar un módulo de búsqueda y navegación que utilice la tecnología de Alfresco Community 5.2.

Palabras claves: buscador, documentos digitales, repositorio digital, obtención de documentos.

## Abstract

XABAL Repxos 3.0 is a system for the implantation of digital repositories in scientific institutions and Academic, developed in the Documentation Management Computerization Center, belonging to the University of Informatics Sciences. All the information generated in the different types of events that take place in the university, is stored in XABAL Repxos; its integration with the investigative processes, allows to use the documentation as a source of information, offering a way for the dissemination of knowledge and also a tool in the teaching-educational process.

The large amount of information stored in XABAL Repxos requires the existence of a search engine to obtain the desired results in the shortest time possible, this repository is based on the technology provided by DSpace, which includes its own search engine to satisfy this need, as an initiative we want to change all the technology of the project to Alfresco Community 5.2 which is a newer technology, it is simpler, more powerful and facilitates the relationship between all the components of the different products.

Due to this existing situation it was decided to develop a search and navigation module that uses Alfresco Community 5.2 technology.

**Keywords:** search engine, digital documents, digital repository, obtaining documents

# Índice

Introducción .....	<b>Error! Bookmark not defined.</b>
Capítulo 1: Fundamentación teórica de la investigación .....	<b>Error! Bookmark not defined.</b>
1.1 Introducción del capítulo .....	<b>Error! Bookmark not defined.</b>
1.2 Conceptos asociados .....	<b>Error! Bookmark not defined.</b>
1.3 Ejemplos de aplicaciones .....	<b>Error! Bookmark not defined.</b>
<b>Resultados de los estudios de otras aplicaciones</b> .....	<b>Error! Bookmark not defined.</b>
1.4 Metodologías, tecnologías, herramientas y lenguajes utilizados .....	<b>Error! Bookmark not defined.</b>
<b>1.4.1 Metodología</b> .....	<b>Error! Bookmark not defined.</b>
<b>1.4.2 Herramientas y lenguajes utilizados</b> .....	<b>Error! Bookmark not defined.</b>
1.5 Conclusiones del capítulo .....	<b>Error! Bookmark not defined.</b>
Capítulo 2: Análisis y Diseño de la propuesta de solución.....	<b>Error! Bookmark not defined.</b>
2.1 Introducción del capítulo .....	<b>Error! Bookmark not defined.</b>
2.2 propuesta de solución .....	<b>Error! Bookmark not defined.</b>
2.3 Modelo de dominio.....	<b>Error! Bookmark not defined.</b>
2.4 Requisitos.....	<b>Error! Bookmark not defined.</b>
<b>2.4.1 Requisitos funcionales</b> .....	<b>Error! Bookmark not defined.</b>
<b>2.4.2 Requisitos no funcionales</b> .....	<b>Error! Bookmark not defined.</b>
2.5 Casos de Uso .....	<b>Error! Bookmark not defined.</b>
2.6 Diagrama de casos de uso del sistema.....	<b>Error! Bookmark not defined.</b>
<b>2.6.1 Descripción de los casos de uso del sistema</b> .....	<b>Error! Bookmark not defined.</b>
2.7 Conclusiones del capítulo .....	<b>Error! Bookmark not defined.</b>
Capítulo 3: Diseño, implementación y pruebas de software. ....	<b>Error! Bookmark not defined.</b>
3.1 Introducción al capítulo .....	<b>Error! Bookmark not defined.</b>
3.1 Patrón Arquitectónico .....	<b>Error! Bookmark not defined.</b>
3.3 Patrones de diseño utilizado en el desarrollo del módulo .....	<b>Error! Bookmark not defined.</b>
3.4 Estándares de codificación.....	<b>Error! Bookmark not defined.</b>
3.5 Diagrama de despliegue .....	<b>Error! Bookmark not defined.</b>



3.6 Pruebas de software.....	<b>Error! Bookmark not defined.</b>
3.7 Métodos seleccionados.....	<b>Error! Bookmark not defined.</b>
<b>3.7.1 Método de prueba caja blanca</b> .....	<b>Error! Bookmark not defined.</b>
<b>3.7.2 Método de prueba caja negra</b> .....	<b>Error! Bookmark not defined.</b>
3.8 Conclusiones del capítulo .....	<b>Error! Bookmark not defined.</b>
Conclusiones .....	<b>Error! Bookmark not defined.</b>
Referencias Bibliográficas .....	<b>Error! Bookmark not defined.</b>
Bibliografía.....	<b>Error! Bookmark not defined.</b>
Anexos.....	<b>Error! Bookmark not defined.</b>

## Índice de tablas

Tabla 1 Descripción del CUS búsqueda simple .....	<b>Error! Bookmark not defined.</b>
Tabla 2 Descripción del CUS búsqueda avanzada .....	<b>Error! Bookmark not defined.</b>
Tabla 3. Caso de prueba para el camino 1.....	<b>Error! Bookmark not defined.</b>
Tabla 4. Caso de prueba para el método de caja negra .....	<b>Error! Bookmark not defined.</b>
Tabla 5. Resultado de las Pruebas .....	<b>Error! Bookmark not defined.</b>
Tabla 6. Descripción del caso de uso navegar en el sistema .....	<b>Error! Bookmark not defined.</b>
Tabla 7. Descripción del caso de uso mostrar resultados .....	<b>Error! Bookmark not defined.</b>
Tabla 8. Caso de prueba para el camino 2.....	<b>Error! Bookmark not defined.</b>

## Índice de figuras

Ilustración 1. Fases e implementación .....	<b>Error! Bookmark not defined.</b>
Ilustración 2. Diagrama de modelo de dominio .....	<b>Error! Bookmark not defined.</b>
Ilustración 3. Diagrama de casos de uso de sistema .....	<b>Error! Bookmark not defined.</b>
Ilustración 4. Patrón modelo-Vista-Controlador .....	<b>Error! Bookmark not defined.</b>
Ilustración 5. Diagrama de despliegue .....	<b>Error! Bookmark not defined.</b>
Ilustración 6. Código base del método validar formulario .....	<b>Error! Bookmark not defined.</b>
Ilustración 7. Esquema para la técnica del camino básico .....	<b>Error! Bookmark not defined.</b>

## Introducción

El empleo de las Tecnologías de la Información y las Comunicaciones (TIC) es de reconocida importancia en la actualidad. Estas se refieren a todos los instrumentos, procesos y soportes que están destinados a optimizar la comunicación humana. En las empresas, instituciones y organizaciones contar con la información oportuna para tomar las mejores decisiones en el momento adecuado, resulta algo fundamental.

Las empresas e instituciones han logrado informatizar los servicios y el control de documentos de dichos lugares a través de herramientas y nuevas formas de almacenamiento digital, todo esto debido al constante desarrollo de la información, esta se maneja en favor de los objetivos estratégicos de una organización y en el cumplimiento de las misiones encomendadas a cada uno de sus individuos.

En la actualidad los sistemas de gestión documental constituyen una de las alternativas para gestionar grandes volúmenes de documentos e información. Los sistemas de gestión de documentos facilitan la creación, recepción, organización, almacenamiento, seguridad, así como capacidades de recuperación de archivos. Un ejemplo de ello son los repositorios digitales. Un repositorio es un sitio web centralizado donde se mantiene y almacena la información digital en las bases de datos o archivos informáticos. Pueden contener los archivos en su servidor o referenciar desde su web al alojamiento originario. (Repositorios digitales, 2015)

*“Los repositorios son herramientas eficaces para almacenar, organizar y hacer uso eficiente de la información y el conocimiento. Los recursos almacenados en los repositorios deben poseer una estructura estandarizada de metadatos que asegure la accesibilidad de los recursos digitales almacenados en el sistema. Más concretamente, los repositorios digitales son realizados para organizar, gestionar, preservar y dar libre acceso a la producción científica, académica, institucional y de cualquier otro ámbito cultural, generada por los miembros de la institución en soporte digital”.* (Aroca Cámara, Checa Claudel, & Guzmán Pérez)

La Universidad de las Ciencias Informáticas (UCI, por sus siglas en español), surgida el 23 de septiembre de 2002 con la visión de futuro del Comandante en Jefe Fidel Castro Ruz, es actualmente una de las universidades más jóvenes de Cuba. Creada bajo la concepción de ser un centro de nuevo tipo, de alcance nacional, con tareas concretas en el proyecto de informatización de la sociedad y énfasis en el desarrollo de la industria del software. Estas características hacen de la UCI una universidad atípica que, debido a su estructura y características forma una parte indispensable en el proceso de gestión de la información y el conocimiento para las empresas y entidades. Es por ello que el Centro de Informatización de la Gestión Documental (CIGED por sus siglas en español) perteneciente a la Facultad 2 desarrolló XABAL Repxos (Sistema para Repositorios Institucionales) en su versión 3.0. Este sistema gestiona todas las producciones

científicas e investigativas generadas en la universidad con el objetivo de mantenerlas centralizadas y disponibles.

Permite el almacenamiento, la visualización y la obtención de ítems (documento digital con sus metadatos asociados) mediante búsquedas simples y avanzadas. Este es un sistema web que permite almacenar y difundir la producción intelectual digital generada por una institución. Basado en la herramienta Dspace (software de código abierto que provee herramientas para la administración de colecciones digitales) la cual es ampliamente utilizada como base para el desarrollo de estos repositorios como es el caso de repositorios conocidos internacionalmente como Acceda, Aquatic Commons y Social Science Research Network, tiene como objetivos principales garantizar la visibilidad de los autores, facilitar el contacto entre ellos, favorecer la discusión de los trabajos depositados, contribuir al aumento de las citas y al impacto de los trabajos en la comunidad científica de la institución donde se instale.

Toda la información generada en los distintos tipos de eventos que se desarrollan en la institución, es almacenada en XABAL REPXOS 3.0; su integración con los procesos investigativos, permite utilizar la documentación como fuente de información, ofreciendo una vía para la divulgación del conocimiento y también una herramienta en el proceso docente-educativo. Se facilita de esta forma el acceso abierto a la información a los investigadores de la institución.

La gran cantidad de información almacenada en XABAL Repxos hace necesario la existencia de un buscador para obtener los resultados deseados en el menor tiempo posible, este repositorio está montado sobre la tecnología que brinda Dspace, el cual incluye su propio buscador, pero este solo permite visualizar los documentos en formato pdf y doc imposibilitando la obtención de documentos en cualquier otro formato por eso se quiere integrar la tecnología del proyecto a Alfresco que es una tecnología más reciente, más sencilla, más potente, facilita la relación entre todos los componentes de los distintos productos y posee un potente visualizador capaz de mostrar los documentos de cualquier formato.

Teniendo en cuenta la situación problemática descrita se formula como **problema a resolver**: ¿Cómo integrar el módulo de búsqueda y navegación de XABAL Repxos con tecnología Dspace con Alfresco Community 5.2?

Determinando para la investigación como **Objeto de estudio** los procesos de búsqueda y navegación de documentos en repositorios digitales.

**.El objetivo general de este trabajo es:** Desarrollar el Módulo de búsqueda y navegación de documentos de XABAL Repxos en la tecnología de Alfresco Community 5.2 para integrar la tecnología de búsqueda de Dspace con Alfresco, enmarcado en el **campo de acción**: sistemas de recuperación de documentos

**Preguntas científicas:**

1. ¿Cuáles son las tendencias y tecnologías existentes en el mundo, las herramientas, tecnologías, metodología de desarrollo de software y lenguajes de programación a tener en cuenta para abordar la solución del problema planteado?
2. ¿Cómo precisar las características con que contará el módulo de búsqueda y navegación a desarrollar y cómo realizar su análisis y diseño?
3. ¿Cómo desarrollar el módulo de búsqueda y navegación que permita obtener documentos con la tecnología de Alfresco Community 5.2?
4. ¿Cómo determinar el correcto funcionamiento del módulo de búsqueda y navegación de documentos de XABAL Repxos 3.0 con la tecnología de Alfresco Community 5.2?

**Tareas de investigación:**

1. Revisión bibliográfica para conformar el estado del arte de la investigación y selección de las tecnologías, metodologías y herramientas a utilizar en el proceso de desarrollo a partir de la revisión de las más utilizadas.
2. Descripción de la propuesta de solución para determinar los requisitos funcionales y no funcionales que darán paso al diseño del módulo.
3. Desarrollo del módulo de búsqueda y navegación que permita obtener documentos con la tecnología de Alfresco Community 5.2
4. Validación mediante la aplicación de pruebas de software a la solución propuesta.

Los métodos científicos a utilizar en la investigación estuvieron determinados por el objetivo general y las tareas de investigación previstas.

**Métodos teóricos:**

**Analítico–sintético:** para realizar un estudio bibliográfico de la teoría existente alrededor del objeto de estudio, determinar las características que tendrá la propuesta de solución y definir las tecnologías y herramientas a utilizar para el desarrollo de la propuesta de solución. Para dividir el objeto de estudio en conceptos que serán examinados por separado, estudiándose rigurosamente cada uno de ellos de manera independiente y confrontando el criterio de disímiles autores y las correspondencias entre ellos.

**Inductivo-deductivo:** para analizar una muestra representativa de trabajos asociados al desarrollo de software vinculado a XABAL Repxos, encontrando las particularidades y elementos en común entre ellos, para posteriormente extrapolar este conocimiento a una propuesta que permita dar solución a la problemática planteada.

**Histórico-lógico:** con la utilización de este método se podrá realizar el estudio del estado del arte e investigar acerca de aplicaciones o soluciones similares, en este caso se contará como fuente principal de estudio el Gestor de contenido empresarial Dspace.

**Modelación:** para representar la solución a través de esquemas y diagramas que apoyen la documentación del sistema.

#### **Métodos Empíricos:**

**Observación:** Se empleó para evaluar los resultados obtenidos contra los resultados esperados con el desarrollo del módulo.

**Entrevista:** se realiza una entrevista a los especialistas del proyecto Repositorio Institucional, mediante la cual se obtiene toda la información posible para el desarrollo del Módulo de búsqueda y navegación del sistema XABAL Repxos, procesos de negocio, requisitos funcionales y no funcionales. [Ver anexo 3](#)

Para una mejor comprensión de la investigación, se decidió definir una estructura capitular que aporte el grado de organización necesario y facilite la lectura del documento. Estará compuesto por tres capítulos:

Capítulo 1: Fundamentación teórica de la investigación. Expone los elementos teóricos que sustentan la investigación. Se realiza un estudio de las soluciones similares y se analizan las metodologías, tecnologías y herramientas que se ajustan al desarrollo de la investigación, justificando la selección y utilización de cada una de ellas.

Capítulo 2: Análisis y Diseño de la propuesta de solución. Se describe la propuesta de solución, se muestran los artefactos correspondientes a las dos primeras fases de la metodología de desarrollo seleccionada y se muestran los prototipos del sistema.

Capítulo 3: Implementación y pruebas de software de la solución. Se muestran los artefactos generados en el diseño del módulo, se describen los patrones arquitectónicos utilizados. Se describe el proceso de pruebas realizadas a la propuesta de solución, con el fin de verificar la calidad de los datos almacenados y cumplir las expectativas del cliente. Se documentaron los resultados a cada una de las pruebas realizadas.

Se muestran los artefactos correspondientes a las dos últimas fases de la metodología de desarrollo seleccionada. Se describe el proceso de diseño e implementación del repositorio digital, obteniéndose el modelo lógico, las tablas de hechos y dimensiones, y las uniones entre estas.



# Capítulo 1: Fundamentación teórica de la investigación

## 1.1 Introducción del capítulo

En el presente capítulo se analizan las tendencias y tecnologías existentes en el mundo, relacionadas con los repositorios digitales, además de las definiciones de los términos a utilizar para una mejor comprensión del contexto del problema. Se fundamenta el uso de las herramientas, tecnologías, metodología de desarrollo de software y lenguajes de programación que se utilizan, lo que constituye el basamento teórico de la investigación.

## 1.2 Conceptos asociados

### 1.2.1 Repositorio

Un repositorio es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos. Los datos almacenados en un repositorio pueden distribuirse a través de una red informática, como Internet, o de un medio físico, como un disco compacto. Pueden ser de acceso público o estar protegidos y necesitar de una autenticación previa. Los repositorios más conocidos son los de carácter académico e institucional. Los repositorios suelen contar con sistemas de respaldo y mantenimiento preventivo y correctivo, lo que hace que la información se pueda recuperar en el caso que la máquina quede inutilizable. A esto se lo conoce como preservación digital. (De Giusti, Marisa Raquel; Lira, Ariel Jorge; Villarreal, Gonzalo Luján; Texier, José ,noviembre de 2012)

### 1.2.2 Metadato

*“(...) es toda aquella información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato u objeto que tiene la finalidad de facilitar su recuperación, autenticación, evaluación y preservación (...)”* (José A. Senso, Antonio de la Rosa Piñero. Scielo Scientific Electronic Library Online)

### 1.2.3 Ítem

En DSpace ítem se refiere a un fichero en sí, conocido como “bitstreams” y el conjunto de metadatos que describe ese fichero. (Equipo de desarrollo de DSpace 5.x Documentation [En línea] 2015)

### 1.2.4 XABAL Repxos 3.0

Es un sistema para la implantación de repositorios digitales. Permite adaptarse a múltiples usos para la gestión y recuperación de documentos digitales en cualquier formato. Permite incorporar y buscar documentos además de navegar dentro de ellos, administrar el sistema y realizar suscripciones a las diferentes colecciones. Está basado en el software DSpace. (Informáticas(UCI), Universidad de las Ciencias, 2015.)

**1.2.5 Formulario:** Es un conjunto de campos solicitados por un determinado programa que los almacenará y que serán utilizados o bien manipulados, en caso de ser necesario ejercer algún tipo de modificación sobre ellos. (Definición abc. [En línea] [Citado el: 04 de mayo de 2019.]

**1.2.6 Documento Digital:** un documento que no recibe su nombre de la forma mediática en que se presenta la información, sino directamente de la manera particular en que se registra. En estos la información, no importa su forma: texto, sonido, imagen fija o en movimiento, se registra en un medio. (Gastaminza, Félix del Valle, 2006.)

### 1.3 Ejemplos de aplicaciones

La necesidad cada vez más creciente de almacenar el gran cúmulo de información que se ha recopilado año tras año ha sido un factor favorable para el desarrollo de nuevos repositorios digitales en todo el mundo. Existen una variedad de ejemplos de repositorios científicos a lo largo de todo el mundo, sobre los que se hará alusión a continuación:

**PubMed Central:** Se trata de un repositorio temático con un enorme éxito lo cual ha dado pie a que se haya creado una red llamada PMI International con el objeto de establecer archivos abiertos en colaboración con instituciones locales de cada país. Se encuentra especializado en Medicina. Se ha convertido en una fuente de referencia para investigadores de todo el mundo. Para realizar la búsqueda este sistema utiliza Gerión que es una herramienta de descubrimiento que te permite realizar búsquedas desde un sólo punto de acceso y de forma integrada a toda la colección de la BV -SSPA (bases de datos, otros recursos, revistas-e y libros-e) de reconocido prestigio y calidad científica, tanto suscrito por la Biblioteca Virtual, como de acceso libre vía Internet.

**DSpace@Cambridge:** Se trata de un servicio de la Universidad de Cambridge gestionado por la biblioteca y el servicio de informática. Ofrece artículos, tesis e informes técnicos en diferentes formatos. Su objetivo es preservar y difundir los materiales digitales relacionados con la Universidad. Este para realizar la búsqueda utiliza la forma que propone Dspace, que ofrece realizarla en todo el sistema, o bien restringirla a una Comunidad o Colección específica, cambiar el número de resultados que aparezcan en cada página y ordenar dichos resultados por relevancia, título, fecha de publicación o fecha de envío, tanto en orden descendente como en orden ascendente.

Para las búsquedas de múltiples términos, el motor de búsqueda de DSpace utiliza por defecto el operador booleano OR, que requiere que al menos uno de dichos términos esté presente para concretar una respuesta.

**Social Science Research Network (SSRN):** Se trata de un repositorio en línea de investigación académica dentro de las Ciencias Sociales y Humanidades. Permite compartir de una forma rápida y eficaz los trabajos entre autores y lectores. Su objetivo es difundir la investigación en Ciencias Sociales. Este también realiza la búsqueda basado en Dspace explicado anteriormente.

**AQUATIC COMMONS:** se trata de un repositorio digital que cubre temas ambientes marinos naturales, estuarios de agua salobre y de agua fresca. Incluye todos los aspectos de la ciencia, tecnología, administración y conservación de estos ambientes, sus organismos y recursos, y los aspectos económicos, sociológicos y legales. Este realiza la búsqueda la búsqueda con BASE (del acrónimo inglés Bielefeld Academic Search Engine) que es un motor de búsqueda multidisciplinario, que es usado generalmente para encontrar textos en recursos escolares a través de Internet. Este buscador fue creado por la Biblioteca de la Universidad de Bielefeld en la ciudad de Bielefeld, Alemania. Se basa en la tecnología de búsqueda provista por la empresa noruega Fast Search & Transfer.

**Acceda:** es el repositorio digital de la Universidad de Las Palmas de Gran Canaria (ULPGC) basado en la tecnología DSpace. Fue creado en octubre de 2009 por la Biblioteca universitaria para recoger la producción científica, en abierto, generada por la comunidad de la ULPGC: docentes, investigadores, estudiantes y personal de administración y servicios. Acceda[21] es una plataforma OAI-PMH que sigue las especificaciones Open AIRE, Hispana y Driver y las políticas de Acceso abierto. También incluye la aplicación BUSstreaming que convierte los vídeos y audios a Flash y los ofrece en streaming y cuenta con un sistema de auto publicación. En 2014, superaba los 8000 documentos entre tesis, trabajos de grado y postgrado, revistas y material didáctico para la investigación. (Acceda: documentación científica de la ULPGC en abierto. 2014). Este también realiza las búsquedas con Dspace.

**En América latina:**

**Red de repositorios latinoamericanos:** Esta red tiene como objetivo proporcionar una herramienta de fácil acceso a las publicaciones electrónicas en texto completo en diferentes repositorios latinoamericanos.

Permite realizar búsquedas simultáneas a través de una sola interfaz, en 75 universidades de Argentina, Brasil, Chile, Colombia, Costa Rica, Ecuador, El Salvador, Guyana, Honduras, México, Perú, Puerto Rico, Trinidad y Tobago y Uruguay. (Universidad de Chile. Sistema de Servicios de Información y Bibliotecas,2013)

**SeDiCI:** es el repositorio Institucional de la Universidad Nacional de La Plata. El repositorio se creó en el 2003 para albergar, preservar y difundir, a través de su sitio web, las creaciones y producciones intelectuales, científicas y artísticas de los diversos actores de la universidad (alumnos, profesores e investigadores). Dado que SeDiCI se adhiere a las políticas de acceso abierto, todo su acervo está disponible de forma libre y gratuita, contando en su gran mayoría con los documentos a texto completo. En el repositorio se incluyen: libros, libros electrónicos, artículos, tesis, tesinas, obras artísticas, revistas, reseñas y documentos legales, entre otros. (SEDICI, 2013). Este utiliza como base para la búsqueda el Dspace descrito anteriormente.

**Repositorio Digital Institucional José María Rosa:** El Repositorio Digital Institucional José María Rosa es el repositorio institucional de la Universidad Nacional de Lanús. Fue creado el 21 de diciembre de 2011 para promover y difundir la producción científica y académica de sus investigadores, profesores, becarios y estudiantes, estableciendo el acceso abierto como política de apoyo e incentivo. El repositorio está integrado por diferentes colecciones: trabajos finales de grado y de posgrado, artículos, libros, partes de libros, presentaciones a congresos, documentos de trabajo, entre otros. La Dirección de Biblioteca y Servicios de Información Documental es la unidad responsable de organizar y dar tratamiento documental a estas colecciones. (Universidad Nacional de Lanús. Consejo Superior, 2011)

#### **En Cuba:**

**Repositorio de Información de Medio Ambiente de Cuba:** Iniciado en junio del 2015, su finalidad es ofrecer acceso abierto a la producción científica e institucional de las Instituciones, grupos de trabajo, investigadores, docentes y personas interesadas en publicar y compartir en la Web la producción científica e institucional relacionada con temas ambientales, cubanos preferentemente; además de garantizar el depósito de las publicaciones de las revistas alojadas en el Portal de Revistas de Medio Ambiente de Cuba.( Cejas Rodríguez, Francisco, Rodríguez Quintana, Mei Emi, Cantillo Cantera, Denis, 2018)

**Repositorio de Tesis en Ciencias Biomédicas y de la Salud:** Es un depósito de tesis en formato digital, cuyo objetivo es archivar, organizar, preservar y difundir las tesis de doctorado suscribiéndose a la iniciativa para el Acceso Abierto a la investigación. Aborda además información de otras ramas afines que estén

relacionadas o respalden las temáticas del Sistema Nacional de Salud de Cuba. (SÁNCHEZ REMÓN, D., REMÓN, D.S. y TARRAGÓ, N.S., 2013)

**Repositorio Institucional del Ministerio de Educación Superior RIMES:** Disponible en la Web el Repositorio Institucional del Ministerio de Educación Superior. RIMES contiene libros publicados por la Editorial Universitaria del Ministerio de Educación Superior (MES) y las Tesis de Doctorado aprobadas por la Comisión Nacional de Grados Científicos de la República de Cuba para su divulgación en la Web. (Chapman Baró, Yarenia de la Caridad, Torricella Morales, Raúl Gonzalo, 2016)

### **Resultado del estudio realizado.**

El análisis realizado a los diferentes repositorios digitales que poseen un módulo de búsqueda permitió conocer características que estos tienen en común y la tecnología q estos utilizan con el objetivo de tenerlos presentes en el desarrollo de esta investigación. Entre las características a tener en cuenta en el módulo a desarrollar se encuentran:

1. Tecnología utilizada
2. Búsqueda
3. Navegación

## 1.4 Metodologías, tecnologías, herramientas y lenguajes utilizados

### **1.4.1 Metodología**

Para el desarrollo de esta investigación se selecciona como metodología a AUP-UCI por ser una variación desarrollada en la universidad adaptada a las diferentes características de los proyectos que se desarrollan en la institución.

### **Fases Variación AUP-UCI (Rodríguez Sánchez, Tamara. 2015.)**

La Metodología Variación de AUP para la UCI está formada por tres fases, (Inicio, Ejecución y Cierre) para el ciclo de vida de los proyectos de la universidad, las cuales contienen las características de las cuatro fases (Inicio, Elaboración, Construcción y Transición) propuestas en AUP. Las características de las fases de la metodología de la universidad son:

**Inicio:** Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto. (Rodríguez Sánchez, Tamara. 2015.)

**Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software. (Rodríguez Sánchez, Tamara. 2015.)

**Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto. (Rodríguez Sánchez, Tamara. 2015.)

### **Disciplinas de la variación AUP-UCI**

#### **Modelado de negocio:**

El Modelado del Negocio es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para modelar el negocio se proponen las siguientes variantes:

1. Casos de Uso del Negocio (CUN)
2. Descripción de Proceso de Negocio (DPN)
3. Modelo Conceptual (MC)

A partir de las variantes anteriores se condicionan cuatro escenarios para modelar el sistema en la disciplina Requisitos. (Rodríguez Sánchez, Tamara. 2015.)

#### **Requisitos:**

El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto. Existen tres formas de encapsular los requisitos [Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP)], agrupados en cuatro escenarios condicionados por el Modelado de negocio. (Rodríguez Sánchez, Tamara. 2015.)

#### **Análisis y diseño:**

En esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a

la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis. (Rodríguez Sánchez, Tamara. 2015.)

### **Implementación:**

En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema. (Rodríguez Sánchez, Tamara. 2015.)

**Pruebas internas:** En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible, componentes de prueba ejecutables para automatizar las pruebas. (Rodríguez Sánchez, Tamara. 2015.)

**Pruebas de liberación:** Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación. (Rodríguez Sánchez, Tamara. 2015.)

**Pruebas de Aceptación:** Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. (Rodríguez Sánchez, Tamara. 2015.) Ver figura 1.

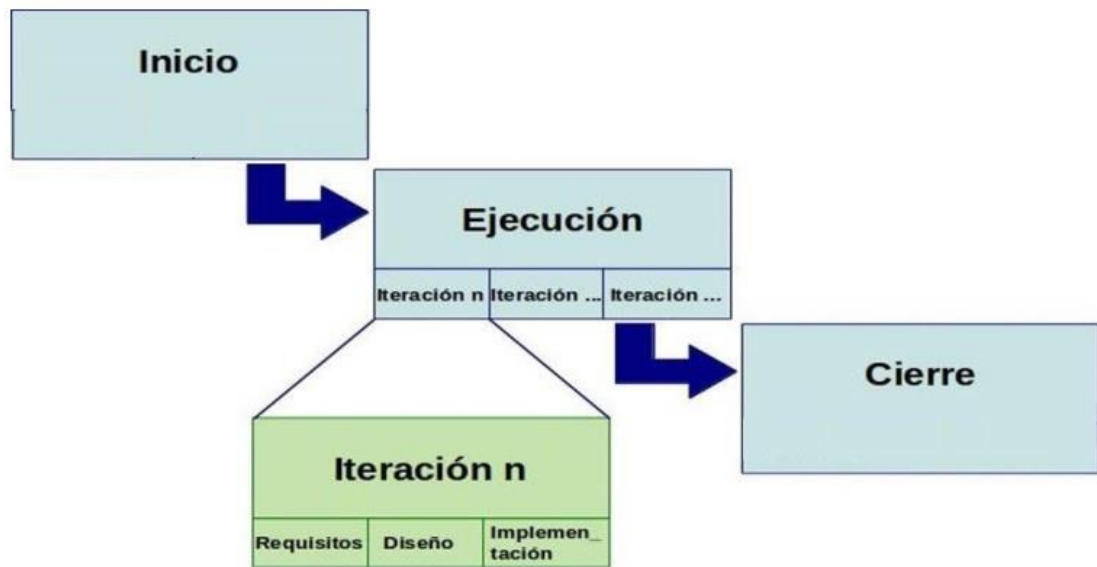


Ilustración 1. Fases e implementación

Esta metodología está compuesta por cuatro escenarios (Rodríguez Sánchez, Tamara. 2015. Metodología de desarrollo para la Actividad productiva de la UCI. La Habana: s.n., 2015.):

Escenario No 1: Proyectos que modelen el negocio con casos de uso del negocio solo pueden modelar el sistema con casos de uso del sistema.

Escenario No 2: Proyectos que modelen el negocio con modelo conceptual solo pueden modelar el sistema con casos de uso del sistema.

Escenario No 3: Proyectos que modelen el negocio con descripción de procesos de negocio solo pueden modelar el sistema con descripción de requisitos por procesos.

Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con historias de usuario.

Se selecciona para esta investigación el escenario número dos porque este aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelan exclusivamente los



conceptos fundamentales del negocio. Se utiliza este escenario ya que es recomendado para proyectos donde el objetivo primario es la gestión y presentación de información como es el caso del presente trabajo.

#### 1.4.2 Herramientas y lenguajes utilizados

**Alfresco Community 5.2:** es un sistema de administración de contenidos de código fuente libre, desarrollado en Java, basado en estándares abiertos y de escala empresarial para sistemas operativos tipo Windows, Unix Solaris y algunas versiones de Linux (<https://www.Alfresco.com/products/networks/compare/>).

Alfresco Community 5.2. es utilizado como software de gestión documental para documentos, páginas web, registros, imágenes y desarrollo colaborativo de contenido. Entre sus principales características se encuentra:

- Gestión de documentos
- Visor que soporta cualquier formato de documento
- Soporte de varios idiomas
- Empaquetamiento de aplicación portable
- Soporte multiplataforma (oficialmente Windows, GNU/Linux)
- Interfaz gráfica basada en navegadores de Internet (oficialmente Internet Explorer y Mozilla Firefox)

**Sistema Gestor de Base de Datos:** PostgreSQL 9.4.6: es un sistema de gestión de bases de datos objeto-relacional y de código abierto, distribuido bajo la licencia BSD y con su código fuente disponible libremente. Se ejecuta en los sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc. Presenta documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios. Soporte nativo para los lenguajes: Java, PHP, C, C++, Perl, Python, etc. Soporte de todas las características de una base de datos profesional (triggers, procedimientos almacenados, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas, etc.). (Informáticas, Comunidad de PostgreSQL Universidad de las Ciencias.2016). Se utilizó este sistema porque es el que trae incluido Alfresco Community 5.2..

#### **JavaScript 1.6:**

JavaScript es un lenguaje interpretado con múltiples propósitos, pero se considera hasta ahora como un complemento. No requiere ningún software de servidor por lo tanto es un lenguaje de script del lado del cliente (Gauchat, 2012). Se utilizó este lenguaje Alfresco posee una Interfaz de Programación de Aplicación

(API por sus siglas en inglés) en su versión 5.2 para JavaScript que permite la modificación y creación de nodos, aspectos y propiedades del repositorio de este. A través de esta API los desarrolladores de scripts podrán encontrar nodos, recorrer jerarquías de nodos, realizar búsquedas, examinar y modificar el valor de propiedades y aspectos. También, con los scripts se pueden crear nuevos ficheros, espacios o nodos de cualquier tipo.

**Visual Studio Code versión 1.33.1:** Code es un editor de código fuente desarrollado por Microsoft para Windows , Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. (Visual Studio Code, 2016).

Es compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un idioma dado. Muchas de las características de Visual Studio Code no están expuestas a través de los menús o la interfaz de usuario. Más bien, se accede a través de la paleta de comandos o a través de archivos .json (por ejemplo, preferencias del usuario).

### **Lenguaje Unificado de Modelado (UML 2.1):**

UML es un lenguaje estándar para especificar, visualizar, construir y documentar los artefactos de sistemas informáticos, así como para el modelado de negocios y otros sistemas. (Lenguaje Unificado de Modelado (UML). [En línea] <http://www.uml.org>.)

UML brinda variados elementos de esquematización para representar las diferentes partes de un sistema, de ellos se utilizan en el desarrollo de la aplicación los diagramas de casos de uso, de clases del diseño, de despliegue, de componentes, de colaboración y el resto de los diagramas.

### **Visual Paradigm 8.0**

Es una herramienta profesional que se utiliza para el modelado, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones. Permite dibujar todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación. (Visual Paradigm. [En línea] <http://www.visual-paradigm.com>.)

## **1.5 Conclusiones del capítulo**

Luego de desarrollar el capítulo se ha podido arribar a las siguientes conclusiones parciales:

El estudio de los principales conceptos permitió una mejor comprensión de la presente investigación.

El estudio de sistemas homólogos permitió identificar las características principales de los sistemas de repositorios digitales tanto nacionales como internacionales.

El estudio de las tecnologías, herramientas, lenguajes y metodologías proporcionó los conocimientos necesarios para su utilización en el desarrollo de la solución propuesta.

## **Capítulo 2: Análisis y Diseño de la propuesta de solución.**

### **2.1 Introducción del capítulo**

El presente capítulo tiene como objetivo describir la propuesta de solución, precisar las características con que contará el módulo, evidenciar mediante artefactos ingenieriles la aplicación de la metodología seleccionada, mostrar los requisitos del sistema, tanto funcionales como no funcionales y realizar el prototipado del sistema.

### **2.2 propuesta de solución**

La propuesta de solución es un módulo de búsqueda y navegación que se encarga de realizar búsquedas de documentos digitales para el sistema XABAL Repxos 3.0. El módulo permitirá a todos los usuarios de XABAL Repxos 3.0 obtener los documentos de su preferencia y poder navegar entre las comunidades existentes para poder visualizar colecciones enteras de documento y elegir el de su preferencia.

La comunicación entre el módulo y las bases de datos se realiza a través del estilo de arquitectura REST, consumiendo los servicios que proporciona el servidor de Dspace.

### **2.3 Modelo de dominio**

El modelo de dominio es una representación gráfica de los conceptos claves del negocio y la relación entre ellos. En él se describen las entidades con sus atributos, las relaciones entre ellas y sus restricciones. Se utiliza el término "Dominio" para diferenciarlo del Modelo de Negocio debido a que es un concepto mucho más amplio. Se centra en una parte del negocio específicamente, la relacionada con el ámbito del proyecto, por lo que el término "dominio" representa una parte del "negocio". El Modelo de Dominio permite aumentar la comprensión del problema y contribuir a esclarecer la terminología o nomenclatura del dominio.

Un modelo del dominio es un diccionario visual de términos importantes del dominio y sus acciones, permite descomponer el espacio del problema en unidades comprensibles (conceptos). Se puede ver como una representación de las clases del mundo real que comunica cuales son los términos importantes y como se relacionan entre sí. (Hall, Craig Larman. Prentice, 2003.)

La solución a desarrollar no posee un modelo del negocio bien definido; por ese motivo, mediante el modelo de dominio que se muestra a continuación se logra la comprensión de los conceptos, los cuales son tratados como objetos que pueden ser clases del dominio o eventos, explicándose cada uno de los conceptos para permitir a los involucrados una fácil comprensión de los procesos que se llevarán a cabo luego de la implementación.

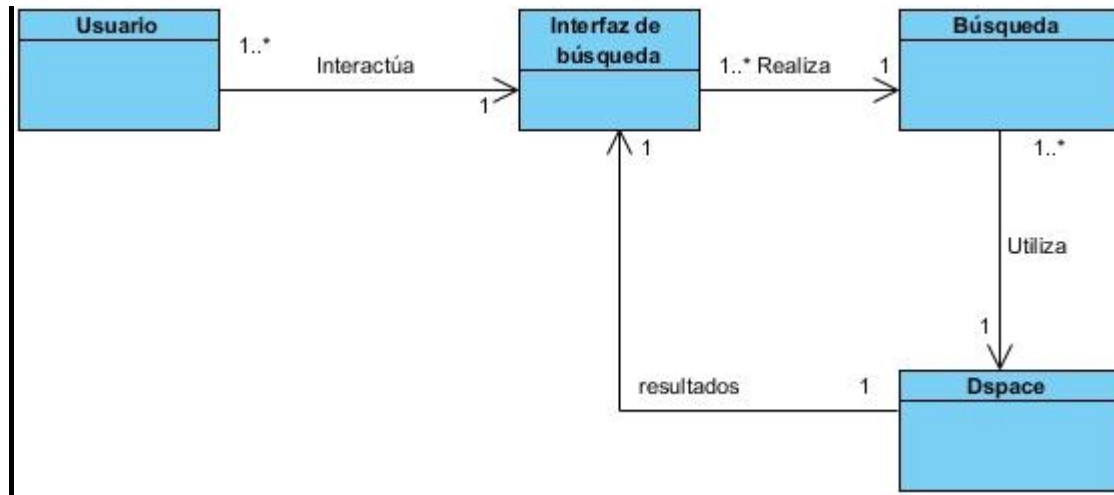


Ilustración 2. Diagrama de modelo de dominio

Los conceptos representados en el modelo de dominio (Fig.1 Diagrama de modelo del dominio) se describen a continuación:

### Diccionario de datos:

**Usuario:** Es el cliente que interactúa con el sistema, una vez registrado y autenticado en él, puede realizar las operaciones disponibles de búsqueda.

**Interfaz de búsqueda:** Es Interfaz en la cual el usuario va a introducir los criterios de búsqueda deseados.

**Búsqueda:** Es el proceso mediante el cual se va a acceder a los documentos archivados en la base de datos.

**Dspace:** Sistema en el cual está basado XABAL Repxos encargado de gestionar todos los documentos en la base de datos.

## 2.4 Requisitos

### 2.4.1 Requisitos funcionales

*“Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. (...) los requerimientos funcionales del sistema describen con detalle la función de éste, sus entradas, salidas y excepciones”.* (Olivera, angel. 2010)

Estos constituyen una forma de representar condiciones o capacidades que debe cumplir un sistema, para satisfacer las necesidades de un contrato, estándar o cualquier documento formal. A continuación, se presenta el listado de las funcionalidades que el sistema debe permitir:

RF1. Cargar comunidades: el módulo debe ser capaz de cargar las comunidades predefinidas en la universidad

RF2. Cargar colecciones: el módulo debe ser capaz de cargar las comunidades predefinidas en la universidad

RF3. Insertar autor: como filtro para realizar una búsqueda avanzada el módulo debe permitir introducir el nombre del autor como parámetro para la búsqueda.

RF4. Insertar fecha de publicación: como filtro para realizar una búsqueda avanzada el módulo debe permitir introducir la fecha de publicación como parámetro para la búsqueda.

RF5. Insertar materia: como filtro para realizar una búsqueda avanzada el módulo debe permitir introducir la materia como parámetro para la búsqueda.

RF6. Insertar título: como filtro para realizar una búsqueda avanzada el módulo debe permitir introducir el título del elemento como parámetro para la búsqueda.

RF7. Mostrar resultados: el módulo debe permitir visualizar los resultados

RF8. Mostrar comunidades: el módulo debe permitir mostrar las comunidades cargadas previamente

RF9. Cargar ítem: el módulo debe cargar los ítems que va a mostrar desde una base de datos remota.

RF10. Mostrar metadatos: cada resultado debe mostrar un metadato asociado.

RF11. Mostrar colecciones: el módulo debe permitir mostrar las colecciones predefinidas por la universidad.

#### **2.4.2 Requisitos no funcionales**

*“Los requerimientos no funcionales rara vez se asocian con características particulares del sistema. Más bien, estos requerimientos especifican o restringen las propiedades o cualidades emergentes del sistema (...).”* (Olivera, angel. 2010)

Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Para el desarrollo de la actual investigación se definen los siguientes requerimientos no funcionales:

**RNF 1:** Usabilidad: El módulo permitirá el fácil acceso a las funcionalidades del mismo. Para ello se escribirá el nombre de la función en cada botón.

**RNF 2:** Hardware: El módulo se desarrolló con los siguientes requerimientos mínimos:

Memoria RAM: 4 GB

Microprocesador: Intel(R) Core(TM) i3-2120

CPU 3.30GHz

Para el correcto funcionamiento del mismo se recomienda su uso en computadoras que tengan las prestaciones mencionadas anteriormente, de lo contrario no se asegura obtener un tiempo de respuesta adecuado.

**RNF 3:** Software: Para acceder al módulo se recomienda utilizar el navegador web Mozilla Firefox, en las versiones superiores a la 33.0, desde los sistemas operativos Windows y GNU/Linux. Además, el navegador debe tener habilitado el uso de JavaScript.

## 2.5 Casos de Uso

Definición de los casos de uso del sistema En correspondencia con los requisitos funcionales del sistema se definieron los siguientes casos de uso:

CU\_1 Búsqueda simple.

CU\_2 Búsqueda avanzada.

CU\_3 Navegar en el sistema.

CU\_4 Mostrar resultados.

## 2.6 Diagrama de casos de uso del sistema

Los diagramas de casos de uso documentan y modelan el comportamiento de un sistema, subsistema o una clase, mostrando las relaciones entre los actores y el caso de uso del sistema.

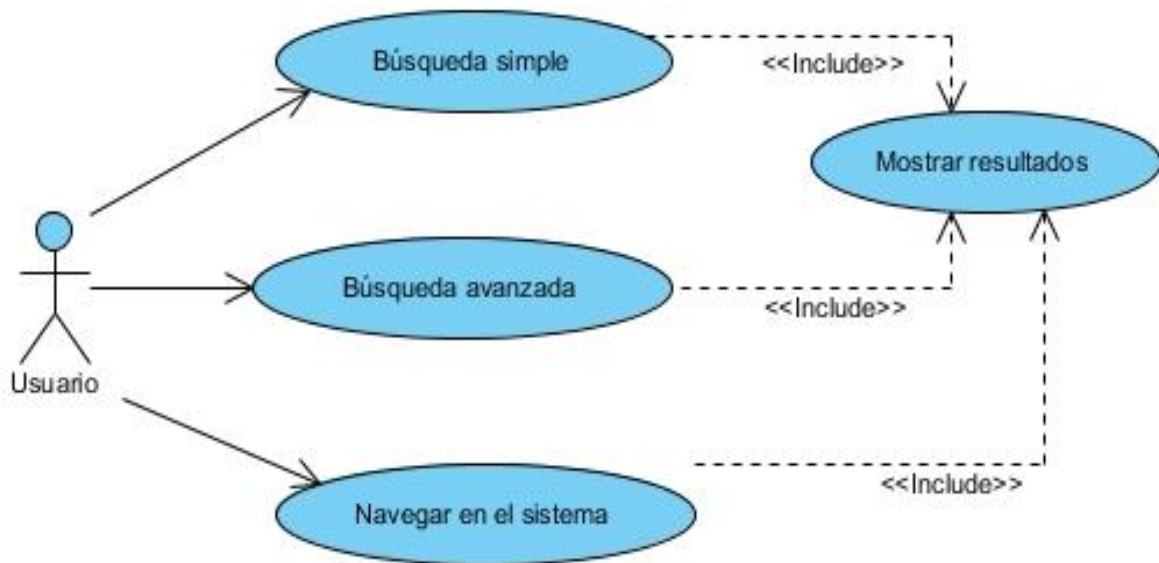


Ilustración 3. Diagrama de casos de uso de sistema

### 2.6.1 Descripción de los casos de uso del sistema

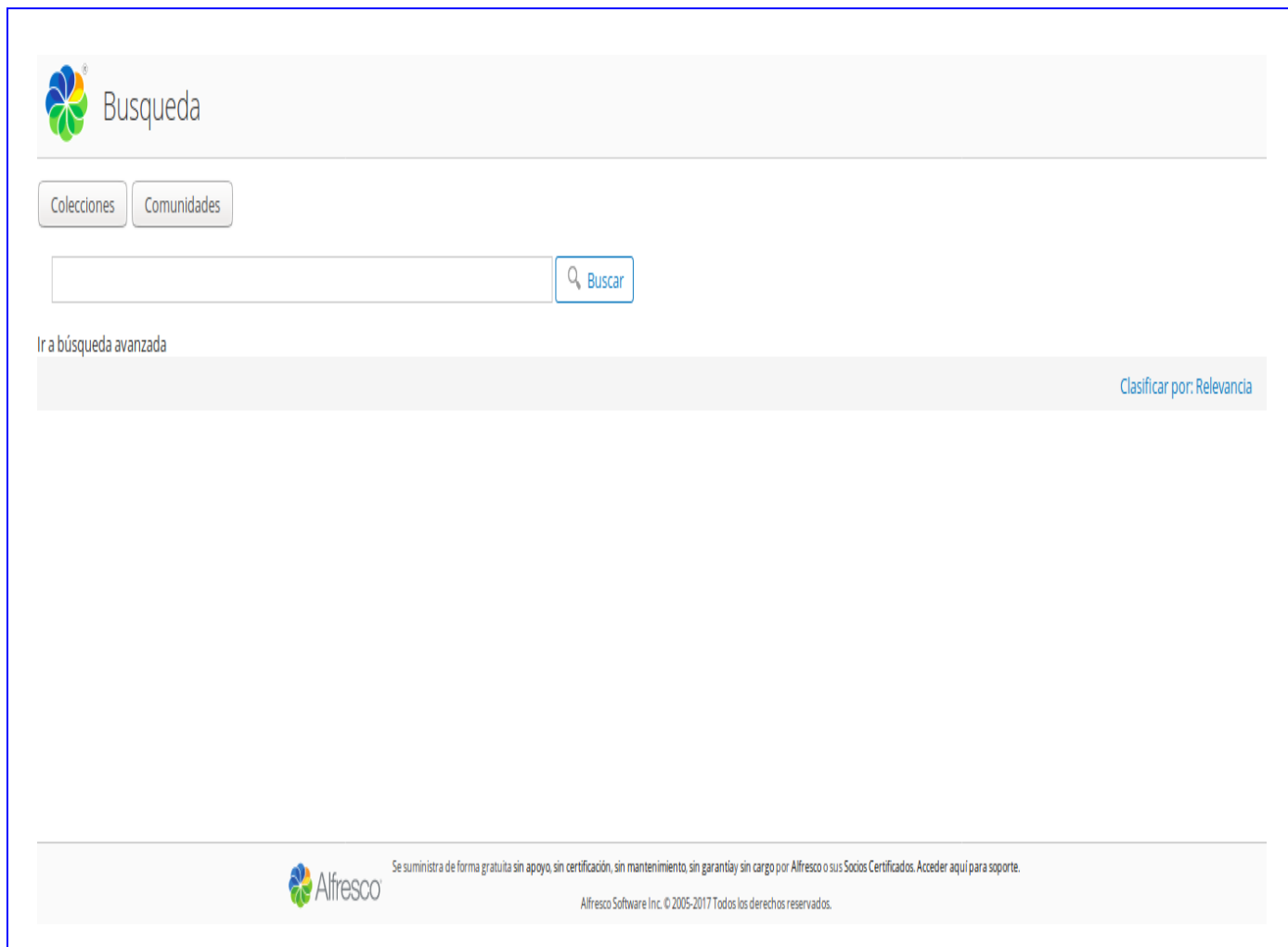
A continuación, se muestra la descripción del caso de uso de sistema Búsqueda simple.

Tabla 1 Descripción del CUS búsqueda simple

<b>Objetivo</b>	Buscar un documento en el repositorio que contenga una expresión entrada por el usuario	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso inicia cuando el usuario necesita buscar un documento en el repositorio	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	Debe existir conectividad con el servidor.	
<b>Postcondiciones</b>	Ninguno o varios ítems fueron encontrados.	
<b>Flujo de eventos</b>		
<b>Flujo básico Búsqueda avanzada</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el cuadro de texto de la búsqueda	
2.		Muestra el cursor de texto parpadeante en forma de línea vertical



3	Escribe la expresión que desea buscar.	
4	Hace click en el botón "Buscar"	
5		Muestra un lista con todos los <i>ítems</i> encontrados.
6		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>4a. No se ha escrito ninguna expresión.</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Comprueba que el cuadro de texto no esté vacío.
2.		Muestra un mensaje: "campos de búsqueda vacío. Inserte criterios de búsqueda".
<b>4ª. Los datos son incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Comprueba que se han introducidos datos incorrectos.
2.		Muestra un mensaje: "Datos incorrectos".
<b>Relaciones</b>	<b>CU incluidos</b>	Mostrar resultados.
	<b>CU extendidos</b>	N/A
<b>Prototipo elemental de interfaz gráfica de usuario</b>		



A continuación, se muestra la descripción del caso de uso de sistema Búsqueda avanzada.


Tabla 2 Descripción del CUS búsqueda avanzada

Objetivo	Realizar búsquedas empleando filtros
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario necesita buscar documentos en el repositorio a través de un autor, fecha, título o materia dado.
Complejidad	Alta
Prioridad	Alta
Precondiciones	Debe existir conexión con el servidor.
Postcondiciones	Ninguno o varios ítems fueron encontrados.

Flujo de eventos		
Flujo básico Búsqueda avanzada		
	Actor	Sistema
1.	Selecciona la opción "Búsqueda avanzada".	
2.	99	Muestra un formulario con los siguientes campos: <ul style="list-style-type: none"> <li>- Autor</li> <li>- Título</li> <li>- Materia</li> <li>- Fecha de modificación</li> </ul> Muestra las siguientes opciones: <ul style="list-style-type: none"> <li>- Aceptar</li> <li>- Cancelar</li> </ul>
3	Hace <i>click</i> en los cuadros de textos de los filtros que desea rellenar.	
4	Hace <i>click</i> en el botón "Buscar"	
5		Muestra una lista con los <i>ítems</i> encontrados
6		Termina el caso de uso
Flujos alternos		
4a. No se ha rellenado ningún campo.		
	Actor	Sistema
3.		Comprueba el valor de los campos
4.		Muestra un mensaje: "Campos vacíos".
4ª. Los datos son incorrectos		
	Actor	Sistema
1.		Comprueba que se han introducidos datos incorrectos.
2.		Muestra un mensaje: "Datos incorrectos".

<b>Relaciones</b>	<b>CU incluidos</b>	Búsqueda avanzada.
	<b>CU extendidos</b>	N/A

**Prototipo elemental de interfaz gráfica de usuario**


Búsqueda avanzada
⏪ Volver a los resulta...

Buscar: Contenido ▾ 🔍 Búsquedas

Palabras clave:

Nombre:

Título:

Descripción:

Tipo MIME:  
Desconocido ▾

Fecha de modificación:

Desde:


DD/MM/AAAA

Hasta:

DD/MM/AAAA

Modificador:

🔍 Búsquedas



Se suministra de forma gratuita sin apoyo, sin certificación, sin mantenimiento, sin garantía y sin cargo por Alfresco o sus Socios Certificados. Acceder aquí para soporte.

Alfresco Software Inc. © 2005-2017 Todos los derechos reservados.

Para más información de la descripción de los CUS consultar el [anexo 1](#).

## 2.7 Conclusiones del capítulo

- Se analizaron y definieron las características de la solución.
- Se especificó el modelo de dominio que permitió definir los conceptos fundamentales del negocio, de esta forma se obtuvo una visión más clara del entorno sobre el cual se desarrolla el problema a resolver.

Al identificar las características con las que debe contar la aplicación se obtuvieron los requisitos funcionales y los requisitos no funcionales.

- Se modeló el diagrama de casos de uso del sistema posibilitando un mejor entendimiento del proceso.
- Con las especificaciones de los casos de uso del sistema se establecieron los flujos básicos de estos y se logró una descripción detallada para una mejor comprensión del sistema que se implementó.

## **Capítulo 3: Diseño, implementación y pruebas de software.**

### 3.1 Introducción al capítulo

El presente capítulo tiene como objetivo detallar la arquitectura a utilizar, los patrones de diseño y la distribución física del sistema mediante el diagrama de despliegue y los diagramas de interacción. También se realizan las pruebas del módulo, evidenciándose la disciplina Pruebas Internas y Pruebas de Aceptación que propone la metodología AUP-UCI para el ciclo de vida del proyecto. En dicho capítulo se diseñan los casos de pruebas para facilitar la disciplina Pruebas internas.

### 3.1 Patrón Arquitectónico

Los patrones arquitectónicos son los relacionados con el diseño a gran escala, que se aplican típicamente durante las primeras iteraciones (la fase de elaboración) cuando se establecen las estructuras y conexiones más importantes. (Larman, Craig, 1999)

El patrón utilizado para la implementación del módulo es el Modelo-Vista-Controlador el cual establece que los componentes de un sistema de software se deben estructurar en tres capas distintas: la capa de acceso a datos, la capa de presentación y la capa de control. De esta manera se engloban los controladores y las vistas en una misma capa, llamada capa web. Además, se introduce una nueva capa, llamada capa de servicios la cual se encarga de implementar la lógica del negocio y de esta forma liberar al controlador de tareas repetitivas, garantizando de esta forma una mayor capacidad de reutilización de código durante el desarrollo.

En el caso más simple, desde una vista el usuario envía los datos que desea procesar en un formulario, estos datos son capturados desde un controlador y enviados a un servicio el cual se encarga de procesarlos y persistirlos en la base de datos, ya sea añadir nuevos datos, consultarlos, editarlos o eliminarlos. Una vez procesados los datos el servicio envía la respuesta al controlador, el cual se encarga de enviarlos a la vista correcta. Por último, la vista es la encargada de dar formato a los datos y mostrarlos al usuario.

Este patrón posee la capacidad de realizar modificaciones en una capa específica sin afectar a las restantes. A continuación, se brinda una descripción detallada de las capas utilizadas para el desarrollo del módulo:

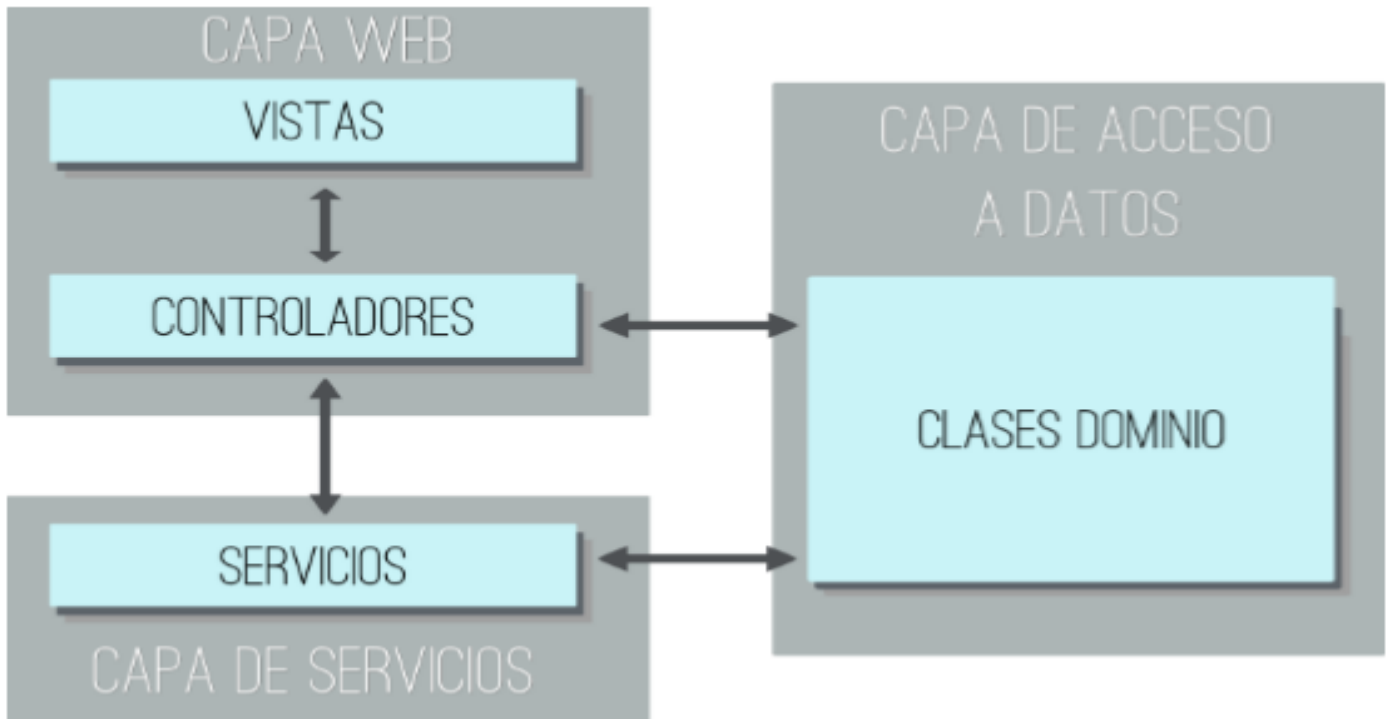


Ilustración 4. Patrón modelo-Vista-Controlador

La capa web está formada por dos partes principales:

**Las vistas:** Son las responsables de presentar la información a los usuarios y están implementadas usando la tecnología Alfresco Community 5.2. Share 5.2.

**Los controladores:** Son los encargados de establecer la comunicación entre las demás capas. Manejan las solicitudes provenientes desde la vista, realizan cambios en el modelo directamente o invocando un servicio y crean la respuesta, la cual puede ser generada en el mismo controlador o la puede delegar a una vista.

La capa de acceso a datos contiene los objetos que representan los datos en la aplicación. Estos pueden ser creados, leídos, actualizados y eliminados y son persistidos en una base de datos para garantizar que perduren en el tiempo. Para evitar el trabajo de forma directa con una base de datos y el uso de consultas SQL, se utiliza GORM, un motor de persistencia construido sobre Hibernate, de esta forma solo es necesario definir clases y relaciones entre ellas y GORM se encarga de realizar todo el trabajo de persistencia, mapeando estas clases en una base de datos.

La capa de servicios es la encargada de implementar la lógica de negocio de la aplicación, dejando a los controladores solo la responsabilidad de manejar el flujo de solicitudes. Por tanto, la función de esta capa

es relacionarse directamente con el modelo de datos, realizar operaciones de lógica de negocio y devolver los datos al controlador.

### 3.3 Patrones de diseño utilizado en el desarrollo del módulo

Un patrón de diseño es una buena práctica documentada de la solución de un problema que ha sido aplicado satisfactoriamente en múltiples entornos. Es una solución recurrente a un problema común observado o descubierto durante el estudio o construcción de numerosas aplicaciones. Su principal objetivo es incrementar la calidad del software en términos de reusabilidad, mantenimiento y extensibilidad. (Larman, Craig, 1999)

#### **Inversión de Control (IoC)**

La inversión de Control es un patrón utilizado, según el cual las dependencias de un componente no deben gestionarse desde el propio componente para que este sólo contenga la lógica necesaria para hacer su trabajo. (Brito, Nacho [En línea], 2009)

Cuando se crea un componente en la aplicación, se gestiona su ciclo de vida (cuándo se crea, cuántas instancias se mantienen vivas a la vez, cómo se destruyen, etc.) y sus dependencias (qué otros componentes se necesitan para realizar su trabajo y cómo conseguirlos). El objetivo de esta técnica es mantener los componentes lo más sencillos que sea posible, incluyendo únicamente código que tenga relación con la lógica de negocio, así la aplicación será más fácil de comprender y mantener. (Brito, Nacho [En línea], 2009)

Este patrón se evidencia en la funcionalidad Comunidades donde solo se declara el servicio que se va a utilizar y no es necesario preocuparse por gestionar dicho servicio ni por las dependencias que este tenga.

#### **Singleton:**

El patrón Singleton asegura que una clase tenga una sola instancia y proporciona un punto de acceso global a ella, de esta forma solo existe una instancia de los servicios durante la ejecución, logrando una reducción de uso de memoria y de tiempo de procesamiento (se elimina la sobrecarga de crear nuevos objetos).

Este patrón se evidencia en el uso del servicio Solr para realizar la búsqueda simple ya que la mayoría de los servicios utilizan este patrón por defecto.

#### **GRASP:**

Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. (Larman, Craig, 1999)



**Experto:** La responsabilidad de realizar una labor específica es otorgada a la clase que contiene la información necesaria para realizar dicha labor. En la propuesta de solución un objeto de tipo File es el responsable de validar sus metadatos y exponer dicha evaluación a los demás objetos, pues la clase File es el experto en esa información. (Larman, 2003)

Este patrón se evidencia en la búsqueda simple cuando se hace uso de la clase Alfresco.search para realizar la búsqueda en vez de utilizar un método en otra clase.

**Bajo Acoplamiento:** El diseño se debe realizar de forma tal que la dependencia entre clases sea baja. De esta forma se tiene clases menos dependientes, logrando una reducción del impacto de los cambios y garantizando un mayor grado de reutilización. Este patrón se pone en práctica al asociar a cada controlador un solo servicio, disminuyendo de esta forma el acoplamiento.

Las funcionalidades del módulo fueron implementadas de forma independiente, esto disminuye la cantidad de cambios a realizar tras hacer una modificación.

**Alta Cohesión:** Los elementos deben tener responsabilidades altamente relacionadas y efectuar la menor cantidad de operaciones posibles. Al existir una alta cohesión se hace más fácil el mantenimiento, entendimiento y la reutilización. La utilización de este patrón se evidencia en los controladores, estos mantienen su código lo más simple posible manejando solo los eventos del sistema y delegando las demás tareas, como la implementación de la lógica de negocio a los servicios.

**Controlador:** Un Controlador es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema. Un Controlador define el método para la operación del sistema. El controlador recibe la solicitud del servicio desde la capa de presentación y coordina su realización, normalmente delegando a otros objetos.

Este patrón se evidencia en el fichero busqueda.js que se encarga controlar muchas de las acciones que se realizan en el módulo.

### 3.4 Estándares de codificación

Los estándares de codificación se definen con vistas a mejorar el entendimiento del código perteneciente al módulo por otros desarrolladores y alcanzar una uniformidad en el mismo. A continuación, se listan los elementos pertenecientes al estándar de codificación definido para el desarrollo del módulo:

- Utilizar nombres en inglés para las clases y métodos.

- Dejar espacio de separación dentro del código, para que sea entendible.
- Escribir en inglés los nombres de las variables.
- Los nombres de las variables deben ser cortos y significativos.

### 3.5 Diagrama de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. (Sparx Systems. [En línea], [Citado el: 13 de marzo de 2019].)

A continuación, se muestra el diagrama de despliegue del módulo para el sistema XABAL Repxos 3.0:

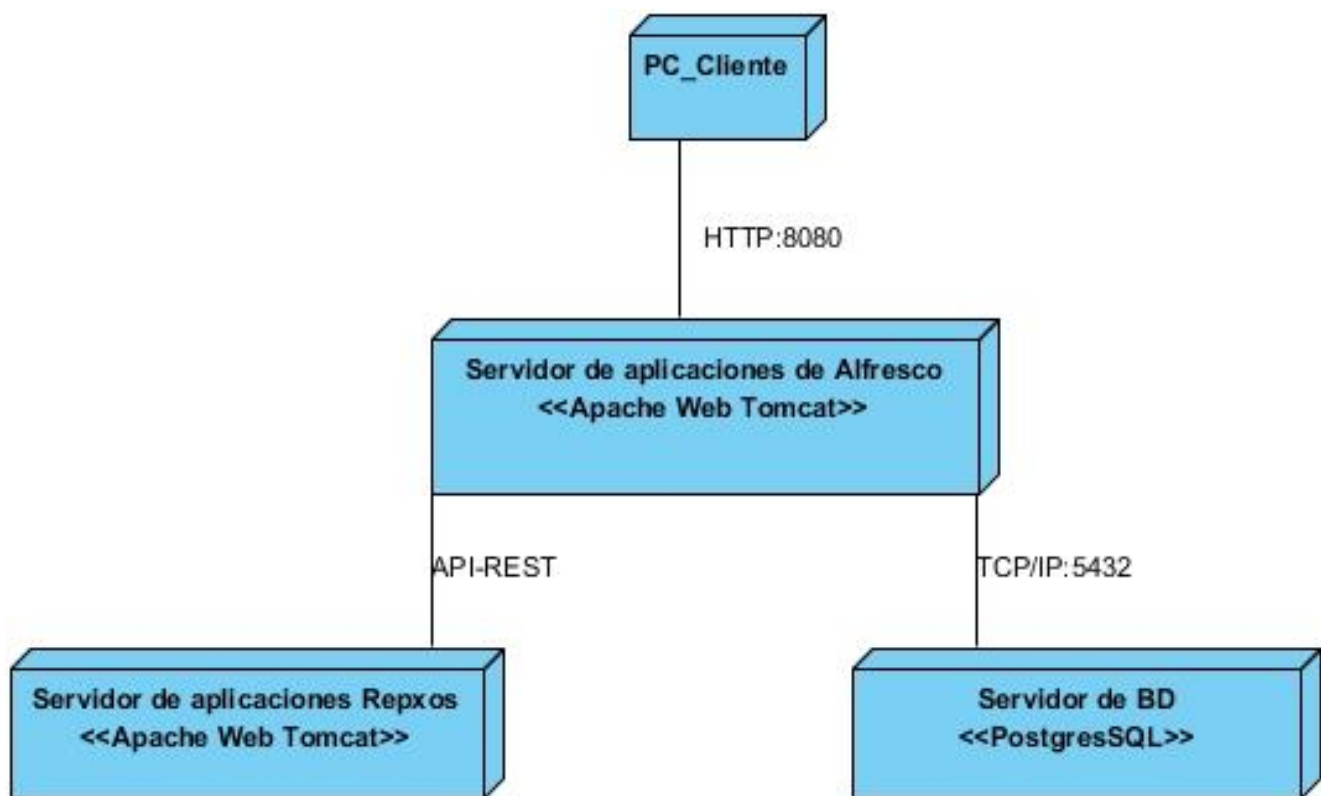


Ilustración 5. Diagrama de despliegue

**PC cliente:** Es el nodo que contiene los navegadores web (Firefox, Chrome, etc.) para acceder a las páginas cliente que interactúan con las páginas servidoras. Se procesan todas las interfaces de usuario que establecen la comunicación entre el usuario y el Módulo Recomendaciones.

**Servidor de aplicaciones Alfresco:** Este nodo tiene alojado Alfresco Community Edition 5.2 que es donde se construye la aplicación, hosteado en un servidor web Apache, y usa la API-REST del sistema XABAL Repxos para capturar los ítems del repositorio.

**Servidor de aplicaciones XABAL Repxos:** En este se encuentra desplegado el sistema XABAL Repxos para poder acceder a la API-REST.

**Servidor de base de datos:** En este servidor es donde se encuentra el sistema gestor de bases de datos PostgreSQL. Aquí se encuentra toda la información estructurada.

En el diagrama de despliegue antes visto se utilizaron los protocolos de comunicación TCP/IP y HTTP debido a que este módulo se desarrolla sobre tecnología web. Se describe a continuación cada uno de los protocolos utilizados:

**Familia de Protocolos TCP/IP:** se utiliza para enlazar computadoras que usan sistemas operativos similares o diferentes, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN). En el caso de este módulo y el sistema XABAL REPXOS se utiliza para interconectarlos con el servidor de base de datos.

**HTTP:** protocolo de transferencia de hipertexto, es usado en cada transacción de la web. Define la sintaxis y semántica que utilizan los elementos software de la arquitectura web (cliente, servidor, proxy para comunicarse). En el módulo se utiliza este protocolo para el acceso de los clientes web a los servicios que brinda la aplicación.

### 3.6 Pruebas de software

Un instrumento adecuado para determinar el estatus de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de prueba. (Pruebas de Software. [En línea] [Citado el: 14 de marzo de 2019.]

La prueba de software es imprescindible para garantizar la calidad de la aplicación representando una revisión final de las especificaciones del diseño y del código.

Elementos a tener en cuenta para que una prueba tenga éxito:

- Niveles de prueba
- Tipos de pruebas
- Método de prueba
- Caso de prueba

- Estrategia de prueba

### **Niveles de prueba**

La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Se distinguen los siguientes niveles de pruebas: (Jorin, Ing. Michael González, 2007)

- Pruebas unitarias
- Prueba de integración
- Prueba de sistema
- Prueba de implantación
- Prueba de aceptación
- Prueba de regresión

Se estará trabajando en el nivel de sistema, con el objetivo de comprobar la funcionalidad de todo el sistema y en el nivel de pruebas unitarias con el objetivo de probar el comportamiento de cada uno de los componentes de forma independiente.

### **Tipos de prueba** (Jorin, Ing. Michael González, 2007)

- Funcionalidad (función, seguridad, volumen, usabilidad)
- Fiabilidad (integridad, estructura, estrés)
- Rendimiento (benchmark, contención, carga, performance profile)
- Soportabilidad (configuración, instalación)

Se aplicaron pruebas de funcionalidad, y dentro de ellas de función. Estas pruebas fijan su atención en la validación de las funciones, métodos, servicios y casos de uso, permitiendo comprobar el correcto funcionamiento de los requisitos funcionales del módulo. Se realizaron además pruebas de rendimiento específicamente de carga, usada para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante.

### **Casos de prueba**

Los casos de prueba deben diseñarse para descubrir errores debidos a cálculos erróneos, comparaciones incorrectas o flujo de control inadecuado. Se diseñan para garantizar que: se satisfagan todos los requisitos de funcionamiento, se logran todas las características de comportamiento, todo el contenido es preciso y se presenta de manera adecuada, se logran todos los requisitos de rendimiento, la documentación es correcta y se satisfacen la facilidad de uso y otros requisitos. (Pressman, Roger S., © 2010)

## Estrategia de prueba

- Describe el enfoque y los objetivos generales de las actividades de prueba.
- Incluye los niveles de prueba a ser diseccionados, el tipo de prueba a ser ejecutada y los casos de prueba diseñados para lograr los objetivos.
- Define:
  - ✓ Técnicas de pruebas (manual o automática) y herramientas a ser usadas.
  - ✓ Criterios de éxitos y culminación de las pruebas.
  - ✓ Consideraciones especiales relacionadas con los recursos necesarios para realizar las pruebas

Una estrategia de prueba del software integra los métodos de diseño de caso de pruebas del software en una serie bien planeada de pasos que desembocará en la eficaz construcción del software. La estrategia proporciona un mapa que describe los pasos que se darán como parte de la prueba, indica cuando se planean y cuando se dan estos pasos, además de cuanto esfuerzo, tiempo y recursos consumirán. Por tanto, cualquier estrategia de prueba debe incorporar la planeación de pruebas, el diseño de casos de pruebas, la ejecución de pruebas y la recolección y evaluación de los datos resultantes. (Pressman, Roger S, 2010)

Una estrategia de prueba del software debe ser lo suficientemente flexible como para promover un enfoque personalizado. Al mismo tiempo, debe ser lo adecuadamente rígido como para promover una planeación razonable y un seguimiento administrativo del avance del proyecto. (Pressman, Roger S, 2010)

## Métodos de Prueba

**Caja negra:** Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requisitos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requisitos funcionales para un programa. (Pressman, Roger S, 2010)

Se basa en técnicas como: Técnica de partición de equivalencia: es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Un caso de prueba ideal descubre de primera mano una clase de errores que de otro modo podrían requerir la ejecución de muchos casos de prueba antes de observar el error general. (Pressman, Roger S, 2010) Técnica de análisis de valores límites: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables. (Jorin, Ing. Michael González, 2007) Técnica de Grafos

de causa-efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones. (Jorin, Ing. Michael González, 2007)

**Caja blanca:** Las pruebas de caja blanca, denominadas a veces pruebas de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba (Pressman, Roger S, 2010). Con la aplicación de este método se garantiza que:

- ✓ Se verifique por lo menos una vez todos los caminos independientes de cada módulo.
- ✓ Se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas.
- ✓ Se ejecuten todos los bucles en sus límites y con sus límites operacionales.
- ✓ Se entrenen las estructuras internas de datos para asegurar su validez.

### 3.7 Métodos seleccionados

La metodología de software empleada en esta investigación define tres disciplinas de pruebas: Pruebas internas, Pruebas de liberación y Pruebas de aceptación. Para dicha investigación se desarrollan las pruebas internas.

#### **Pruebas internas**

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componente de prueba ejecutable para automatizar las pruebas. (Pressman, Roger S, 2010)

Se realizan pruebas al módulo mediante el empleo del método de prueba caja blanca haciendo uso de la técnica asociada a este método, prueba del camino básico y del método de prueba caja negra utilizando la técnica partición equivalente, asociada al mismo.

#### **3.7.1 Método de prueba caja blanca**

Técnica de prueba para el método de caja blanca: Prueba del Camino Básico.

La prueba del camino básico es una técnica de prueba de la Caja Blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. (Pressman, Roger S, 2010)

La técnica consiste en derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se

construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguen para aplicar esta técnica son:

1. A partir del diseño o del código fuente, obtiene el grafo de flujo asociado.

```
function validarFormulario(){  
  
    var txtAutor = document.getElementById('nombre').value;  
    var txtTitulo = document.getElementById('titulo').value;  
    var txtMateria = document.getElementById('materia').value;  
  
    //Test campo obligatorio  
    if(txtNombre == null || txtNombre.length == 0 || /^\\s+$  
$.test(txtNombre)){  
        alert('ERROR: El campo nombre no debe ir vacío o lleno de  
solamente espacios en blanco');  
        return false;  
    }  
}
```

Ilustración 6. Código base del método validar formulario

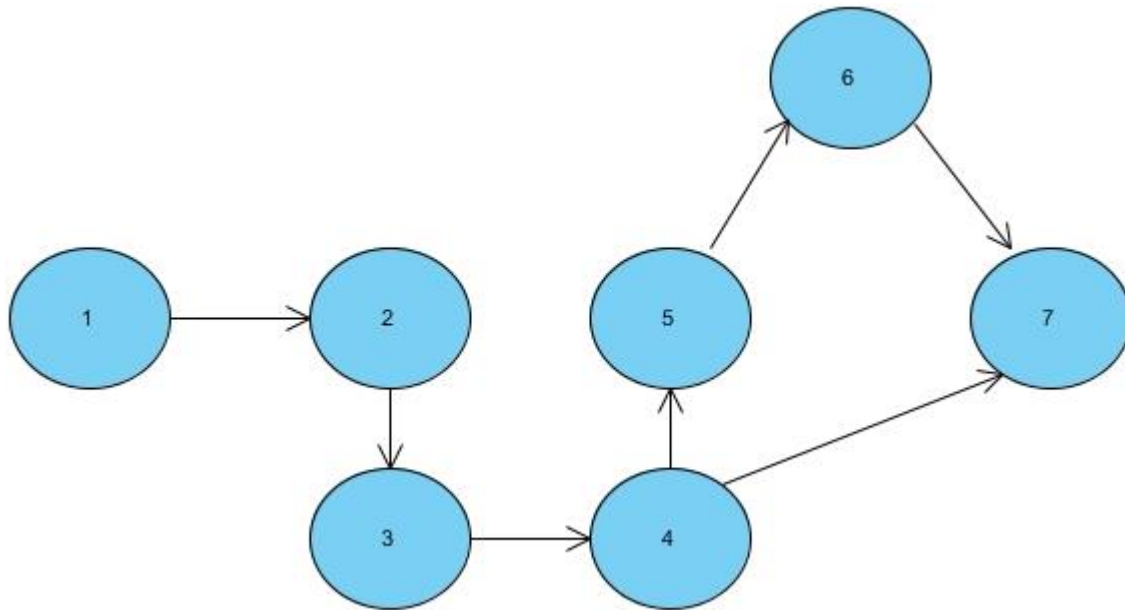


Ilustración 7. Esquema para la técnica del camino básico

2. Se calcula la complejidad ciclomática del grafo obtenido.

$$V(G) = \# \text{ de regiones} = 2$$

$$V(G) = A - N + 2 = 7 - 7 + 2 = 2$$

$$V(G) = P + 1 = 1 + 1$$

3. Se determina un conjunto básico de caminos independientes.

**Primer Camino:** 1-2-3-4-5-6-7

**Segundo Camino:** 1-2-3-4-7

4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Caso de prueba para el camino 1

Tabla 3. Caso de prueba para el camino 1

Descripción	Permite comprobar que los campos no contengan solo espacios.
Condición de ejecución	Los campos autor, título y materia deben contener espacio.
Entrada	Espacios en el nombre del autor, en el título y en la materia.
Resultado	Devuelve un mensaje de error

Ver el caso de prueba para el camino 2 en el [anexo 2](#)

### 3.7.2 Método de prueba caja negra

Técnica de prueba para el método de caja negra: Partición Equivalente.

Esta técnica intenta dividir el dominio de entrada de un programa en un número finito de clases de equivalencia y definir el menor número de casos de prueba que descubran clases de errores. (Sommerville, Ian, 2016)

El diseño de casos de prueba según esta técnica consta de dos pasos:

1. Identificar las clases de equivalencia.

2. Identificar los casos de prueba.

- Una prueba realizada con un valor representativo de cada clase es equivalente a una prueba realizada con cualquier otro valor de dicha clase.
- Si el caso de prueba correspondiente a una clase de equivalencia detecta un error, el resto de los casos de prueba de dicha clase de equivalencia deben detectar el mismo error.



En la siguiente tabla se evidencia el caso de prueba, con los distintos escenarios, su descripción, el ejemplo, la respuesta del sistema y el flujo central, ver ejemplo en la tabla:

Tabla 4. Caso de prueba para el método de caja negra

Escenario	Descripción	Nombre del documento	Respuesta del módulo	Flujo Central
EC 1.1 Buscar documento	El escenario de prueba permite al usuario buscar un documento	V	El módulo verifica los datos y devuelve la lista de ítems que coinciden.	1- Seleccionar el cuadro de búsqueda 2- Insertar nombre del documento a buscar 3- Presionar "Buscar"
		Módulo de referencias para XABAL Repxos		
		Papa		
EC 1.2 Datos incorrectos	El escenario de prueba permite al usuario validar campos incorrectos	I	El módulo muestra un mensaje: "El valor entrado contiene caracteres no validos".	1-Seleccionar el cuadro de búsqueda 2-Insertar nombre del documento a buscar 3-Presionar "Buscar"
		i"#\$		
EC 1.2 Datos incorrectos	El escenario de prueba permite al usuario validar campos incorrectos	I	El módulo muestra un mensaje: "El campo de búsqueda está vacío".	1-Seleccionar el cuadro de búsqueda 2-Insertar nombre del documento a buscar 3-Presionar "Buscar"
EC 1.4 No existe el documento	El escenario de prueba permite al usuario validar la existencia de	V	El módulo muestra un mensaje: "El documento con ese	1-Seleccionar el cuadro de búsqueda
		Aviones rotos		

	documentos con ese nombre.		nombre no está en la base de datos".	2-Insertar nombre del documento a buscar  3-Presionar "Buscar"
--	----------------------------	--	--------------------------------------	--

### Resultado de las pruebas

En la siguiente tabla se exponen los resultados de las pruebas funcionales realizadas al módulo, se desarrollaron 3 iteraciones. Se encontraron 7 no conformidades (NC), en la primera iteración, de ellas 2 fueron errores ortográficos, 4 errores de concordancia entre la aplicación y el diseño de los casos de prueba y 1 error de validación en los datos de entrada, esta última no procede porque el módulo detectó como error un número en el título de un documento y esto no es un error debido a que en un título pueden aparecer números. Se corrigieron las NC detectadas.

En la segunda iteración fueron detectados 2 errores, 1 de ortografía y otro de correspondencia entre el sistema y la descripción de un caso de prueba. Se corrigieron las NC detectadas. Al realizar la tercera iteración, no se detectaron NC, quedando demostrado que los errores encontrados fueron resueltos en su totalidad.

Tabla 5. Resultado de las Pruebas

No. de Iteraciones	Total de no conformidades	No Procede	Resueltas
1ra Iteración	7	1	6
2da Iteración	2	0	2
3ra Iteración	0	0	0

### 3.8 Conclusiones del capítulo

- La arquitectura de software empleada, el diagrama de despliegue y los patrones de diseño sentaron las bases para la construcción del módulo.
- Se definió además los estándares de codificación y las pruebas de software.
- En las pruebas funcionales se encontraron en la primera iteración 7 NC, 1 no procedió y se resolvieron 6, en la segunda iteración se encontraron 2 y se resolvieron ambas y en la tercera no se encontraron NC

## **Conclusiones**

Como resultado de la investigación realizada se desarrolló un módulo de búsqueda y navegación de documentos de XABAL Repxos con la tecnología de Alfresco Community 5.2., concluyendo:

- ✓ Se revisó la bibliografía para conformar el estado del arte de la investigación y se seleccionaron las tecnologías, metodologías y herramientas a utilizar en el proceso de desarrollo a partir de la revisión de las más utilizadas.
- ✓ Se describió la propuesta de solución y se determinaron los requisitos funcionales y no funcionales que dieron paso al diseño del módulo.
- ✓ Se desarrolló el módulo de búsqueda y navegación que permite recuperar documentos con la tecnología de Alfresco Community 5.2
- ✓ la aplicación de pruebas de caja negra y caja blanca a la propuesta de solución propició la correcta validación de sus funcionalidades considerándose lista para ser desplegada en un entorno real.

## Referencias Bibliográficas

1. Hall, Craig Larman. Prentice. 2003. UML y Patrones. 2da Edición. 2003.
2. Scribd. [En línea]. Olivera, angel. 2010. 2010.
3. De Giusti, Marisa Raquel; Lira, Ariel Jorge; Villarreal, Gonzalo Luján; Texier, José (noviembre de 2012). HYPERLINK "<http://sedici.unlp.edu.ar/handle/10915/26045>" Las actividades y el planeamiento de la preservación en un repositorio institucional
4. García-Peñalvo, F. J.; García-Holgado, A. Proceso. Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso, 2017, vol. 2018.
5. José A. Senso, Antonio de la Rosa Piñero. Scielo (Scientific Electronic Library Online). [En línea] [http://www.scielo.br/pdf/ci/v32n2/17038.pdf/..](http://www.scielo.br/pdf/ci/v32n2/17038.pdf/))
6. Informáticas(UCI), Universidad de las Ciencias. Catálogo productos y servicios. La Habana: s.n., 2015.
7. DSpace, Equipo de desarrollo de DSpace 5.x Documentation [En línea] 2015. <https://wiki.duraspace.org/display/DSDOC5x>
8. Universidad de Chile. Sistema de Servicios de Información y Bibliotecas. Red de repositorios latinoamericanos.
9. Universidad Nacional de Lanús. Consejo Superior. 2011. HYPERLINK "[http://www.unla.edu.ar/resoluciones/2011/Diciembre/R.CS.N\\_206-11--21.12.11\\_Reso\\_proyecto\\_aprobado\\_Repositorio\\_Institucional-1.pdf](http://www.unla.edu.ar/resoluciones/2011/Diciembre/R.CS.N_206-11--21.12.11_Reso_proyecto_aprobado_Repositorio_Institucional-1.pdf)" Resolución nº 206/11 .
10. SEDICI. 2013. HYPERLINK "<http://sedici.unlp.edu.ar/blog/2013/11/14/la-ley-de-acceso-abierto-a-la-informacion-cientifica-ya-es-una-realidad/>" La Ley de Acceso Abierto a la Información Científica ya es una realidad .
11. Acceda: documentación científica de la ULPGC en abierto. 2014. Más de 8000 documentos.
12. Rodríguez Sánchez, Tamara. Metodología de desarrollo para la Actividad productiva de la UCI. La Habana: s.n., 2015.
13. Informáticas, Comunidad de PostgreSQL Universidad de las Ciencias. PostgreSQL Comunidad Técnica de Desarrollo. [En línea] 2019. [https://postgresql.uci.cu/?page\\_id=30](https://postgresql.uci.cu/?page_id=30).
14. NetBeans. [En línea] © 2016. [Citado el: 15 de febrero de 2019.] [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html).

15. Lenguaje Unificado de Modelado (UML). [En línea] <http://www.uml.org>.
16. Visual Paradigm. [En línea] <http://www.visual-paradigm.com>.
17. GROUSSARD, T., 2014. JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). S.l.: Ediciones ENI. ISBN 978-2-7460-9347-8.
18. Hibernate | Marco de Desarrollo de la Junta de Andalucía. [en línea], [sin fecha]. [Consulta: 23 enero 2019]. Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/97>.
19. Larman, Craig. UML y patrones. Introducción al análisis y diseño orientado a objetos. Mexico : Prentice Hall, 1999 . ISBN 970-17-0261-1.
20. ooCities. [En línea] [Citado el: 10 de marzo de 2019.] <http://www.oocities.org/es/monsalvelaura/fase2/analisis.html#5>.
21. Brito, Nacho. Manual de desarrollo web con GRAILS. [En línea] 9 de junio de 2009. [Citado el: 13de marzo de 2019.] <http://www.manual-de-grails.es>.
22. Pressman, Roger S. INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO. Séptima edición. Impreso en México: MCGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V, © 2010. ISBN: 978607-15-0314-5.
23. Ivar Jacobson, Grady Booch y James Rumbaugh. El Proceso Unificado De Desarrollo de Software. Madrid: Addison Wesley, 2000. ISBN 84-7829-036-2.
24. Sparx Systems. [En línea] 2000-2016. [Citado el: 13 de marzo de 2019.] [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_deploymentdiagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html).
25. Jorin, Ing.Michaell González. Tesis de maestría. Proceso de pruebas para la liberación de productos software. Ciudad de La Habana: s.n., 2007
26. <https://www.Alfresco Community 5.2..com/products/networks/compare/>
27. Language Support in Visual Studio Code. Visual Studio Code. 2016
28. Sommerville, Ian. Software Engineering. Essex : 10th, 2016.
29. Definición abc. [En línea] [Citado el: 04 de mayo de 2019.] <http://www.definicionabc.com/general/formulario.php>.

30. GASTAMINZA, Félix del Valle. Documentos digitales: hipertexto y edición digital. Universidad de Complutense Madrid. 2006.
31. Villa Ortiz, Freddy Javier, Torres Camba, Belgica Graciela, 2015. Estudio, configuración y puesta en producción de un sistema gestor documental Alfresco Community 5.2. basado en open source. Universidad Politécnica Salesiana. 2015.
32. Cejas Rodríguez, Francisco, Rodríguez Quintana, Mei Emi, Cantillo Cantera, Denis. Repositorio de Información de Medio Ambiente de Cuba: una herramienta para la divulgación científica. [www.revistas.geotech.cu/index.php/abc](http://www.revistas.geotech.cu/index.php/abc).2018.
33. SÁNCHEZ REMÓN, D., REMÓN, D.S. y TARRAGÓ, N.S., [sin fecha]. Repositorio de Tesis Doctorales para el Sistema Nacional de Salud de Cuba. Social Science Libraries in Action [en línea], [Consulta: 11 junio 2019]. Disponible en: [https://www.academia.edu/15446019/Repositorio\\_de\\_Tesis\\_Doctorales\\_para\\_el\\_Sistema\\_Nacional\\_de\\_Salud\\_de\\_Cuba](https://www.academia.edu/15446019/Repositorio_de_Tesis_Doctorales_para_el_Sistema_Nacional_de_Salud_de_Cuba).
34. Chapman Baró, Yarenia de la Caridad, Torricella Morales, Raúl Gonzalo. Repositorio Institucional del Ministerio de Educación Superior (RIMES). 2016.
35. GAUCHAT, J.D., 2012. *El gran libro de HTML5, CSS3 y Javascript*. Primera. Barcelona: MARCOMBO, S.A. ISBN 978-84-267-1782-.

## Bibliografía

1. Alfresco Community 5.2. software y servicios | ECM | BPM. [en línea], [sin fecha]. [Consulta: 13 noviembre 2018]. Disponible en: <https://www.Alfresco Community 5.2..com/es/>.
2. 12 características de Java. 12Características [en línea], 2018. [Consulta: 23 enero 2019]. Disponible en: <https://www.12caracteristicas.com/java/>.
3. BHANDARI, A., CHOUDHARY, V. y MAJMUDAR, P., 2012. Alfresco Community 5.2. Share. S.I.: Packt Publishing Ltd. ISBN 978-1-84951-711-9.
4. Hibernate | Marco de Desarrollo de la Junta de Andalucía. [en línea], [sin fecha]. [Consulta: 23 enero 2019]. Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/97>.
5. JOSE ANTONIO ALBALAT ALMENARA, 13:21:24 UTC. Interfaz Alfresco Community 5.2.. [en línea]. S.I. [Consulta: 15 enero 2019]. Disponible en: <https://es.slideshare.net/josealbalat/interfaz-Alfresco Community 5.2..>
6. GROUSSARD, T., 2014. JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). S.I.: Ediciones ENI. ISBN 978-2-7460-9347-8.
7. ESPINOSA, G.Y.R., RODRÍGUEZ, G.D., FERNÁNDEZ, L.C.A. y CURIAUT, A.V., 2018. Sistema para Repositorios Institucionales XABAL REPXOS 3.0: ventajas e importancia para la gestión de la información y el conocimiento. Revista Publicando, vol. 5, no. 14, pp. 549-561. ISSN 1390-9304.
8. Hall, Craig Larman. Prentice. 2003. UML y Patrones. 2da Edición. 2003.
9. Scribd. [En línea]. Olivera, angel. 2010. 2010.
10. De Giusti, Marisa Raquel; Lira, Ariel Jorge; Villarreal, Gonzalo Luján; Texier, José (noviembre de 2012). HYPERLINK "<http://sedici.unlp.edu.ar/handle/10915/26045>" Las actividades y el planeamiento de la preservación en un repositorio institucional
11. García-Peñalvo, F. J.; García-Holgado, A. Proceso. Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso, 2017, vol. 2018.



12. José A. Senso, Antonio de la Rosa Piñero. Scielo (Scientific Electronic Library Online). [En línea]<http://www.scielo.br/pdf/ci/v32n2/17038.pdf/..>)
13. Informáticas(UCI), Universidad de las Ciencias. Catálogo productos y servicios. La Habana: s.n., 2015.
14. DSpace, Equipo de desarrollo de DSpace 5.x Documentation [En línea] 2015. <https://wiki.duraspace.org/display/DSDOC5x>
15. Universidad de Chile. Sistema de Servicios de Información y Bibliotecas. Red de repositorios latinoamericanos.
16. Universidad Nacional de Lanús. Consejo Superior. 2011. HYPERLINK "[http://www.unla.edu.ar/resoluciones/2011/Diciembre/R.CS.N\\_206-11--21.12.11\\_Reso\\_proyecto\\_aprobado\\_Repositorio\\_Institucional-1.pdf](http://www.unla.edu.ar/resoluciones/2011/Diciembre/R.CS.N_206-11--21.12.11_Reso_proyecto_aprobado_Repositorio_Institucional-1.pdf)" Resolución nº 206/11 .
17. SEDICI. 2013. HYPERLINK "<http://sedici.unlp.edu.ar/blog/2013/11/14/la-ley-de-acceso-abierto-a-la-informacion-cientifica-ya-es-una-realidad/>" La Ley de Acceso Abierto a la Información Científica ya es una realidad .
18. Acceda: documentación científica de la ULPGC en abierto. 2014. Más de 8000 documentos.
19. Rodríguez Sánchez, Tamara. Metodología de desarrollo para la Actividad productiva de la UCI. La Habana: s.n., 2015.
20. Informáticas, Comunidad de PostgreSQL Universidad de las Ciencias. PostgreSQL Comunidad Técnica de Desarrollo. [En línea] 2019. [https://postgresql.uci.cu/?page\\_id=30](https://postgresql.uci.cu/?page_id=30).
21. NetBeans. [En línea] © 2016. [Citado el: 15 de febrero de 2019.] [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html).
22. Lenguaje Unificado de Modelado (UML). [En línea] <http://www.uml.org>.
23. Visual Paradigm. [En línea] <http://www.visual-paradigm.com>.
24. GROUSSARD, T., 2014. JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). S.I.: Ediciones ENI. ISBN 978-2-7460-9347-8.
25. Hibernate | Marco de Desarrollo de la Junta de Andalucía. [en línea], [sin fecha]. [Consulta: 23 enero 2019]. Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/97>.
26. Larman, Craig. UML y patrones. Introducción al análisis y diseño orientado a objetos. Mexico: Prentice Hall, 1999 . ISBN 970-17-0261-1.

27. ooCities. [En línea] [Citado el: 10 de marzo de 2019.] <http://www.oocities.org/es/monsalvelaura/fase2/analisis.html#5>.
28. Brito, Nacho. Manual de desarrollo web con GRAILS. [En línea] 9 de junio de 2009. [Citado el: 13 de marzo de 2019.] <http://www.manual-de-grails.es>.
29. Pressman, Roger S. INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO. Séptima edición. Impreso en México: MCGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V, © 2010. ISBN: 978607-15-0314-5.
30. Ivar Jacobson, Grady Booch y James Rumbaugh. El Proceso Unificado De Desarrollo de Software. Madrid: Addison Wesley, 2000. ISBN 84-7829-036-2.
31. Sparx Systems. [En línea] 2000-2016. [Citado el: 13 de marzo de 2019.] [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_deploymentdiagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html).
32. Jorrián, Ing. Michael González. Tesis de maestría. Proceso de pruebas para la liberación de productos software. Ciudad de La Habana: s.n., 2007
33. <https://www.AlfrescoCommunity5.2.com/products/networks/compare/>
34. Language Support in Visual Studio Code. Visual Studio Code. 2016
35. Sommerville, Ian. Software Engineering. Essex : 10th, 2016.
36. Definición abc. [En línea] [Citado el: 04 de mayo de 2019.] <http://www.definicionabc.com/general/formulario.php>.)
37. GASTAMINZA, Félix del Valle. Documentos digitales: hipertexto y edición digital. Universidad de Complutense Madrid. 2006.
38. Villa Ortiz, Freddy Javier, Torres Camba, Belgica Graciela, 2015. Estudio, configuración y puesta en producción de un sistema gestor documental Alfresco Community 5.2. basado en open source. Universidad Politécnica Salesiana. 2015.
39. Cejas Rodríguez, Francisco, Rodríguez Quintana, Mei Emi, Cantillo Cantera, Denis. Repositorio de Información de Medio Ambiente de Cuba: una herramienta para la divulgación científica. [www.revistas.geotech.cu/index.php/abc.2018](http://www.revistas.geotech.cu/index.php/abc.2018).
40. SÁNCHEZ REMÓN, D., REMÓN, D.S. y TARRAGÓ, N.S., [sin fecha]. Repositorio de Tesis Doctorales para el Sistema Nacional de Salud de Cuba. Social Science Libraries in Action [en línea], [Consulta: 11 junio 2019]. Disponible en: [https://www.academia.edu/15446019/Repositorio\\_de\\_Tesis\\_Doctorales\\_para\\_el\\_Sistema\\_Nacional\\_de\\_Salud\\_de\\_Cuba](https://www.academia.edu/15446019/Repositorio_de_Tesis_Doctorales_para_el_Sistema_Nacional_de_Salud_de_Cuba).

41. Chapman Baró, Yarenia de la Caridad, Torricella Morales, Raúl Gonzalo. Repositorio Institucional del Ministerio de Educación Superior (RIMES). 2016.
42. GAUCHAT, J.D., 2012. *El gran libro de HTML5, CSS3 y Javascript*. Primera. Barcelona: MARCOMBO, S.A. ISBN 978-84-267-1782-.

## Anexos

### Anexo 1: Descripción de los CUS

A continuación, se muestra la descripción del caso de uso de sistema Navegar en el sistema.

Tabla 6. Descripción del caso de uso navegar en el sistema

<b>Objetivo</b>	Navegar en el sistema																	
<b>Actores</b>	Usuario																	
<b>Resumen</b>	El caso inicia cuando el usuario hace click en la opción el botón “Colecciones” o el botón “Comunidades”, se despliega una lista con los elementos existentes en dependencia de la opción seleccionada, y termina cuando el usuario encuentra el conjunto de ítems que desea obtener.																	
<b>Complejidad</b>	Media																	
<b>Prioridad</b>	Alta																	
<b>Precondiciones</b>	Debe existir conectividad con el servidor.																	
<b>Postcondiciones</b>	Varios ítems fueron encontrados.																	
<b>Flujo de eventos</b>																		
<b>Flujo básico</b>	<b>Búsqueda avanzada</b>																	
	<b>Actor</b>	<b>Sistema</b>																
1.	Hace click en el botón “Colecciones” o en el botón “comunidades” en dependencia de por donde desee navegar el usuario																	
2.		<p>Desplegar las colecciones o las comunidades en dependencia de la opción seleccionada en forma de lista:</p> <table> <tr> <td>Comunidades:</td> <td>Colecciones:</td> </tr> <tr> <td>Facultad1</td> <td>-CDAE</td> </tr> <tr> <td>Facultad2</td> <td>-CESOL</td> </tr> <tr> <td>Facultad3</td> <td>-CIDI</td> </tr> <tr> <td>Facultad4</td> <td>-CISED</td> </tr> <tr> <td>Citec</td> <td>-Departamentos</td> </tr> <tr> <td>Otras Areas</td> <td>docentes de la</td> </tr> <tr> <td>Tesis</td> <td>cultad 1</td> </tr> </table>	Comunidades:	Colecciones:	Facultad1	-CDAE	Facultad2	-CESOL	Facultad3	-CIDI	Facultad4	-CISED	Citec	-Departamentos	Otras Areas	docentes de la	Tesis	cultad 1
Comunidades:	Colecciones:																	
Facultad1	-CDAE																	
Facultad2	-CESOL																	
Facultad3	-CIDI																	
Facultad4	-CISED																	
Citec	-Departamentos																	
Otras Areas	docentes de la																	
Tesis	cultad 1																	

		-Entre otras
3	Selecciona la opción que desea vizualizar	
4		Muestra un lista con todos los <i>ítems</i> encontrados.
5		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>1ª. No existe comunicación con el servidor</b>		
	<b>Actor</b>	<b>Sistema</b>
5.		Comprueba que se tenga conexión con el servidor
6.		Muestra un mensaje: "No existe conexión con el servidor".
<b>Relaciones</b>	<b>CU incluidos</b>	Mostrar resultados.
	<b>CU extendidos</b>	N/A
<b>Prototipo elemental de interfaz gráfica de usuario</b>		



➔ Ir a búsqueda avan...

Colecciones

Comunidades

Buscar

Ir a búsqueda avanzada

4 resultado(s) encontrado(s) en todos los sitios.

Clasificar por: Nombre



### Meeting Notes 2011-01-27.doc

Documento en el sitio Sample: Web Site Design Project de tamaño 72 KB modificado por Mike Jackson modificado Jue 24 Feb 2011 11:16:37

En carpeta: /Meeting Notes



### Meeting Notes 2011-02-03.doc

Documento en el sitio Sample: Web Site Design Project de tamaño 72 KB modificado por Mike Jackson modificado Jue 24 Feb 2011 11:16:37

En carpeta: /Meeting Notes



### Meeting Notes 2011-02-10.doc

Documento en el sitio Sample: Web Site Design Project de tamaño 73 KB modificado por Mike Jackson modificado Jue 24 Feb 2011 11:16:37

En carpeta: /Meeting Notes



### Project Contract.pdf (Project Contract for Green Energy)

Contract for the Green Energy project

Documento en el sitio Sample: Web Site Design Project de tamaño 373 KB modificado por Administrator modificado Mar 14 Jun 2011 06:28:54

En carpeta: /Agency Files/Contracts



Se suministra de forma gratuita sin apoyo, sin certificación, sin mantenimiento, sin garantía y sin cargo por Alfresco o sus Socios Certificados. Acceder aquí para soporte.

Alfresco Software Inc. © 2005-2017 Todos los derechos reservados.

A continuación, se muestra la descripción del caso de uso de sistema Mostrar resultados.

Tabla 7. Descripción del caso de uso mostrar resultados

<b>Objetivo</b>	Mostrar los resultados de la búsqueda y la navegación	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso inicia cuando el usuario hace click en la opción el botón “Buscar” y se muestra una lista con los resultados obtenidos que permite selecciona cualquiera de los <i>Items</i> de esta para mostrar sus metadatos, además permite cargar el <i>ítem</i> seleccionado para su consulta.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	Debe existir conectividad con el servidor. Debe haberse presionado el botón “Buscar” o se debe haber entrado a la navegación del módulo	
<b>Postcondiciones</b>	El ítem seleccionado fue cargado.	
<b>Flujo de eventos</b>		
<b>Flujo básico Búsqueda avanzada</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el botón “Buscar”	
2.		Muestra una lista con los resultados En forma de lista de <i>ítems</i>
3	Selecciona el <i>ítem</i> que desee visualizar	
4		Carga el <i>ítem</i> seleccionado y muestra sus metadatos
	Si los metadatos del <i>ítem</i> son los de su interés, lo selecciona para visualizarlo	
		Muestra el contenido del <i>ítem</i> seleccionado
5		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>1ª. No existe comunicación con el servidor</b>		
	<b>Actor</b>	<b>Sistema</b>

7.		Comprueba que se tenga conexión con el servidor
8.		Muestra un mensaje: "No existe conexión con el servidor".
<b>Relaciones</b>	<b>CU incluidos</b>	Búsqueda avanzada Búsqueda simple Navegar en el sistema
	<b>CU extendidos</b>	N/A
<b>Prototipo elemental de interfaz gráfica de usuario</b>		





Buscar

Ir a búsqueda avan...

Colecciones

Comunidades

budget

Buscar

Ir a búsqueda avanzada

8 resultado(s) encontrado(s) en todos los sitios.

Clasificar por: Nombre



42fcbae6-b1fe-4028-9f85-9ad7f81a8e3b

Elemento de lista de datos en el sitio Sample: Web Site Design Project modificado por Administrator modificado Mar 14 Jun 2011 06:28:57



budget.xls (Web Site Design - Budget)

Budget file for the web site redesign

Documento en el sitio Sample: Web Site Design Project de tamaño 26 KB modificado por Mike Jackson modificado Mar 15 Feb 2011 16:35:26

En carpeta: /Budget Files



budget.xls discussion

Carpeta en el sitio Sample: Web Site Design Project modificado por Mike Jackson modificado Mie 16 Feb 2011 05:30:10

En carpeta: /Budget Files/budget.xls



Budget Files (Project finance files)

This folder holds the project budget and invoices

Carpeta en el sitio Sample: Web Site Design Project modificado por Mike Jackson modificado Mar 15 Feb 2011 16:08:20

En carpeta: /



Meeting Notes 2011-01-27.doc

Documento en el sitio Sample: Web Site Design Project de tamaño 72 KB modificado por Mike Jackson modificado Jue 24 Feb 2011 11:16:37

En carpeta: /Meeting Notes



Meeting Notes 2011-02-10.doc

Documento en el sitio Sample: Web Site Design Project de tamaño 73 KB modificado por Mike Jackson modificado Jue 24 Feb 2011 11:16:37

En carpeta: /Meeting Notes



Meetings

Página wiki en el sitio Sample: Web Site Design Project de tamaño 2 KB modificado por Administrator modificado Mar 14 Jun 2011 06:28:57



Milestones

Página wiki en el sitio Sample: Web Site Design Project de tamaño 3 KB modificado por Administrator modificado Mar 14 Jun 2011 06:28:57



Se suministra de forma gratuita sin apoyo, sin certificación, sin mantenimiento, sin garantía sin cargo por Alfresco o sus Socios Certificados. Acceder aquí para soporte.

Alfresco Software Inc. © 2005-2017 Todos los derechos reservados.

## Anexo 2: Caso de prueba

A continuación, se muestra el caso de prueba para el camino básico número dos de la técnica del camino básico.

Tabla 8. Caso de prueba para el camino 2

Descripción	Permite comprobar que los campos estén llenos correctamente
Condición de ejecución	Los campos autor, título y materia deben estar bien llenados correctamente.
Entrada	Rellenar los campos correctamente
Resultado	Retorna falso

## Anexo 3: Entrevista

La entrevista se realizó de manera informal. Las preguntas fluyeron en dependencia del conocimiento de los especialistas del centro sobre el modulo implementado.

1. ¿Cómo funciona el proceso de búsqueda y navegación en el proyecto de Repositorio Institucional?
2. ¿Qué funcionalidades tiene el módulo de búsqueda y navegación del sistema XABAL Repxos?
3. ¿Cuál es el objetivo de implementar este módulo?
4. ¿Qué datos se manejan en el módulo?
5. ¿Qué requisitos debe tener el módulo a desarrollar para que sea compatible con otros productos del centro?

Entrevista realizada a:

Analista del proyecto Repositorio Institucional: Gleidis Yuriannis Rosabal Espinosa

Desarrollador en el proyecto Repositorio Institucional: José Javier Hernández Benítez

