



Universidad de las Ciencias Informáticas

Facultad 2

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

**Módulo Referencias Bibliográficas del Sistema
para repositorios digitales REPXOS 3.0**

Autores: Alex Ferrer Ladrón de Guevara

Zahylí Rizo Soler

Tutor: Ing. Gleidis Yuriannis Rosabal Espinosa

Co-tutor: Ing. Deylert Pérez Rivera

La Habana, 18 de junio de 2015

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y otorgamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2015.

Autores:

Zahylí Rizo Soler

Alex Ferrer Ladrón de Guevara

Tutores:

Ing. Gleidis Yuriannis Rosabal Espinosa

Ing. Deylert Pérez Rivera

Datos de contacto

Tutor principal:

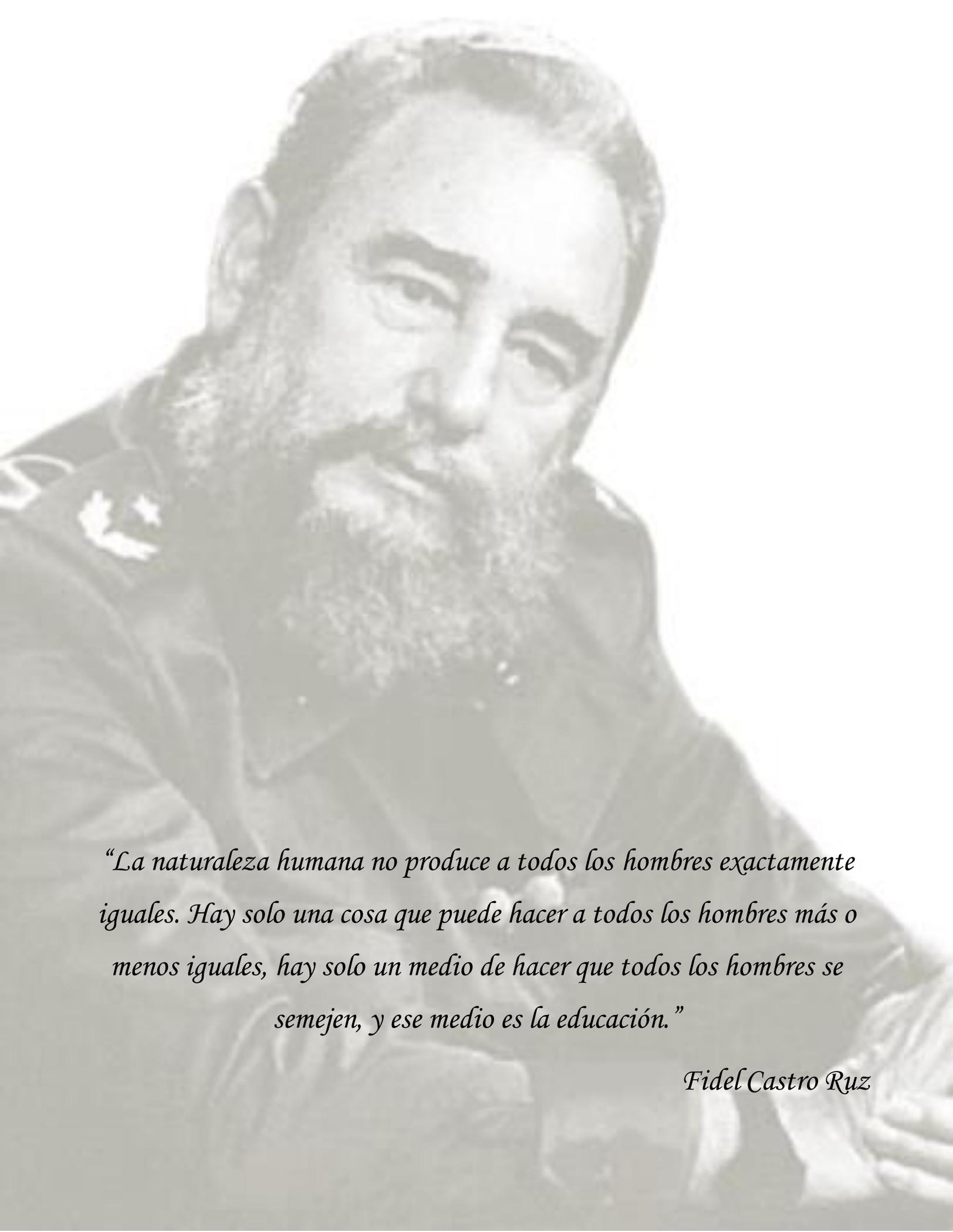
Ing. Gleidis Yuriannis Rosabal Espinosa: graduada en el año 2011 como Ingeniera en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Se desempeña como especialista B en ciencias Informáticas en el Departamento de Aplicaciones del Centro CIGED, ocupando el rol de analista del proyecto Repositorio Institucional.

Correo electrónico: gyrosabal@uci.cu.

Co-tutor:

Ing. Deylert Pérez Rivera: graduado en el año 2013 como Ingeniero en Ciencias Informáticas. Se desempeña como recién graduado en adiestramiento en el Centro CIGED de la Facultad 2, ocupando el rol de desarrollador en el proyecto Repositorio Institucional.

Correo electrónico: drivera@uci.cu.



“La naturaleza humana no produce a todos los hombres exactamente iguales. Hay solo una cosa que puede hacer a todos los hombres más o menos iguales, hay solo un medio de hacer que todos los hombres se semejen, y ese medio es la educación.”

Fidel Castro Ruz

Agradecimientos

Alex

Por contribuir en la formación de la persona que soy hoy, le doy las gracias a:

*Mi **familia**, en especial a mis 3 madres por consagrar su vida a mí en los momentos más difíciles de mi vida, y a mi tío que aun en la distancia me brinda siempre su apoyo y cariño.*

*Los **amigos** del IPI que aun hoy mantienen comunicación conmigo, y que de vez en cuando van a mi casa a jugar DOTA. También a los míos de la UCI que me mantuvieron lo más actualizado posible en temas de series, tecnología entre otros temas.*

*Los **profesores** y **directivos** de la UCI que hicieron posible que hoy se gradúe un profesional con mis características con todas las de la ley.*

*Mis **tutores**, en especial a **Deylert** por madrugar conmigo en ocasiones durante el proceso de desarrollo y por sus exigencias para la conformación de una aplicación con calidad.*

*Mi **compañera** de tesis **Zahylí**, por su entrega para con la realización de esta investigación y los momentos de estrés proporcionados por ella en cada revisión del trabajo realizado.*

*Mis **compañeros** de **clases**, por su tiempo, comprensión y repasos que me dieron en el momento que los necesitaba.*

*Las **pedras** que encontré en el camino, que me hicieron valorar mis actitudes desde otro punto de vista, y que me enseñaron a no conformarme con la victoria de una batalla, sino que es necesario desear más allá del triunfo.*

Agradecimientos

Zahylí

A mis abuelos Pupa y Yaya por quererme más allá de los límites, por sus incansables sacrificios, por consentirme todos mis caprichos, por los valores que me inculcaron y han hecho de mí la mejor persona que puedo ser. No alcanzan las palabras para expresarles cuánto les agradezco. A mi abuela Amalia por cuidar siempre se mí, por su apoyo, sus consejos, por cuidar de mi mamá y mi hermana en mi ausencia y por hacerme siempre sonreír.

A mis padres por su inmenso amor y cariño, por darme siempre las palabras de apoyo cuando las cosas parecen derrumbarse. A mi mamita por ser mi ejemplo de mujer luchadora e inteligente, por hacerme sentir especial, por su dedicación y paciencia, por enseñarme el valor de las cosas, por ser infinitamente mi modelo de persona a seguir. A mi papito por su apoyo en todo momento, por hacerme ver que nunca es tarde para prepararse en la vida, por su perseverancia y por sus consejos. Los amo con todo mi corazón.

A mis hermanos por darme fuerzas para seguir adelante. Para ellos es este triunfo y que les sirva de ejemplo para el resto de sus vidas. A Claudia por ser la mejor hermana del mundo, por cubrirme las espaldas, por hacerme recordar siempre el lado positivo de la vida. A Vale por su cariño a pesar de la distancia, por ser una persona de bien y por cuidarme.

A mis tíos y primos, en especial a Lore, por su apoyo desde lejos, por guiarme siempre por el buen camino, por sus consejos, por mantener siempre unida a la familia más bella del mundo. Los quiero mucho.

A mi novio, “puchito”, por estar siempre a mi lado dándome apoyo, fortaleza, alegrías y espíritus necesarios en los momentos más importantes y difíciles de mi vida, por tu amor incondicional cariño y compañía. Gracias por existir, estar ahí para mí y abrirme las puertas del amor.

A mis amistades Adrián, Cristina, Laurita, Jose, Janet y Neurys que estuvieron siempre que los necesité en lo bueno y en lo malo, por sobrellevarme a pesar de mi carácter, por la confianza depositada en mí, por los consejos y por ser como mi familia, como hermanos más. Espero siempre tenerlos a mi lado

A mis tutores por acogerme con cariño y por brindarme parte de su tiempo a pesar de tanto trabajo. A Gleidis por sus certeras correcciones, por su constante preocupación y dedicación, por hacer los momentos más

estresantes lo más divertido posible y por adoptarme como un pollito más. A Deylert por tantos momentos de estrés y sacrificio que le hicimos pasar, por hacer posible gran parte de este trabajo, por las largas noche programando, por ser un amigo más desde el primer momento.

A mi compañero de tesis Alex, que a pesar de las dificultades por las que la vida lo ha hecho pasar, se ha logrado forjar como un gran profesional, por los momentos de apuro y estrés, por su comprensión y dedicación con el trabajo.

A mi segunda familia, la familia de mi novio que me acogió en su seno y me abrió su corazón para quererme como una hija más.

A todos mis compañeros de aula, a los que perduraron y a los que no por los momentos y experiencias compartidos, por aprender juntos unos de otros

A los profesores por contribuir a formarme profesionalmente y guiarme en todo momento.

A mis compañeros del proyecto que de una forma u otra contribuyeron a la realización de este trabajo, en especial a Luis Carlos, Gleidis, Tan, Bexi, Aramis y Lázaro.

A mi oponente Adrián Gómez García por su exigencia, por hacerme creer en mí y por sus enseñanzas tan valiosas.

A todos muchísimas gracias...

Dedicatoria

Alex

*A mi bisabuela **Marina Rivera**, que en gloria esté observando los pasos de gigante que estoy dando en la vida.*

A mi tío Vladimir, por cumplirme mis sueños tecnológicos dentro de sus posibilidades.

A esas personas que contribuyeron a mi formación como profesional, que despertaron en mí ese bichito de la curiosidad, de saber cada día un poquito más que el anterior.

Zahyli

A Pupa y Yaya, por su dedicación y entrega, a ellos les debo la realización de este trabajo por haber hecho de mi mundo un lugar mejor, como ustedes ninguno. Los quiero mucho

A mi mamá por tanto sacrificio y por ser mi ejemplo ante la vida. Por exigirme siempre por mi superación, para ti es todo mi amor y mis logros en la vida.

A mi papá por creer en mí en todo momento y por inspirarme a realizar mis sueños. Te quiero mucho papito

A mi Claudita que es la luz de mis días, para que se apoye en este triunfo y le sirva de ejemplo en algún momento de su vida.

A mi novio por ser una persona muy importante en mi vida, por estar para mí en los momentos más lindos y en los más difíciles, por aguantar hasta el final para estar conmigo y por hacerme creer que el amor existe de verdad.

A todos lo que confiaron en mí y me apoyaron siempre para la realización de este mi mayor triunfo. Por hacerme mejor cada día.

Resumen

Las referencias bibliográficas son el conjunto de datos y elementos que describen de forma detallada una publicación. La búsqueda y gestión de referencias bibliográficas es una parte fundamental de todo proceso investigativo. El Centro de Informatización de la Gestión Documental (CIGED) de la Universidad de las Ciencias Informáticas (UCI) cuenta con el proyecto Repositorio Institucional que actualmente desarrolla el Sistema para repositorios digitales REPXOS 3.0 que está basado en la herramienta para la creación de repositorios Dspace 4.2. Este sistema brinda un conjunto de servicios de almacenamiento, gestión y distribución de materiales digitales disponibles para los miembros de una determinada institución.

Entre las limitaciones que presenta se encuentra que no permite extraer las referencias bibliográficas de los documentos consultados, sin tener que hacer uso de gestores bibliográficos externos para gestionar correctamente la bibliografía, o realizarlo de forma manual. Para dar solución a este problema se realizó un estudio de sistemas similares que implementan funcionalidades que permitan generar referencias bibliográficas. Se estudiaron los estilos bibliográficos más utilizados actualmente, así como los formatos de archivos más populares en ese ámbito, y se logró conformar con estos conocimientos, una propuesta de solución. Se obtiene como resultado un módulo para el sistema REPXOS 3.0 que permite la extracción y conversión de metadatos de documentos consultados, para la generación de las referencias bibliográficas, además de permitir enviar dichas referencias por correo electrónico a un determinado usuario.

Palabras clave: conversión, estilos bibliográficos, extracción, metadatos, referencias bibliográficas.

Índice

INTRODUCCIÓN	14
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	19
1.1 METADATOS.....	19
1.2 EXTRACCIÓN Y CONVERSIÓN DE METADATOS DE UN DOCUMENTO.....	20
1.2.1. <i>Formatos de archivos más utilizados por gestores bibliográficos.</i>	20
1.3 ESTILOS BIBLIOGRÁFICOS.....	21
1.3.1. <i>Estilos bibliográficos más utilizados para conformar las referencias bibliográficas</i>	22
1.4 ANÁLISIS DE SISTEMAS QUE IMPLEMENTAN COMPONENTES DE SOFTWARE DE REFERENCIAS BIBLIOGRÁFICAS.....	24
1.4.1. <i>Repositorio institucional de la Universidad Oberta de Catalunya (UOC), España</i>	24
1.4.2. <i>Sistema Integrado de Gestión Bibliotecaria (SIGB) de la UCI, Cuba</i>	25
1.4.3. <i>Biblioteca Campus Cuernavaca de la Universidad Tecnológica de Monterrey, México</i>	24
1.4.4. <i>Google Académico</i>	25
1.5 CONCLUSIONES DEL ANÁLISIS DE SISTEMAS SIMILARES	26
1.6 TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL DESARROLLO	26
1.6.1. <i>Metodología de desarrollo de software</i>	26
1.6.2. <i>Plataforma de desarrollo Java Development Kit 7.0 (JDK)</i>	27
1.6.3. <i>Lenguajes de programación</i>	27
1.6.4. <i>Lenguaje Unificado de Modelado (UML)</i>	28
1.6.5. <i>Lenguaje para maquetado CSS 3</i>	28
1.6.6. <i>Herramienta CASE</i>	29
1.6.7. <i>Herramienta de repositorio: DSpace 4.2</i>	29
1.6.8. <i>Herramienta de compilación: Maven 3</i>	29
1.6.9. <i>Servidor Apache Tomcat 7.0</i>	30
1.6.10. <i>Herramienta de construcción de proyectos: Apache Ant 1.8.2</i>	30
1.6.11. <i>Marco de trabajo</i>	30
1.6.12. <i>Sistema Gestor de Base de Datos PostgreSQL 9.3</i>	31
1.6.13. <i>Entorno de Desarrollo Integrado (IDE)</i>	31
CAPÍTULO 2. CARACTERÍSTICAS DEL MÓDULO REFERENCIAS BIBLIOGRÁFICAS	33
2.1 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN	33
2.2 MODELO DE DOMINIO	33

2.2.1.	<i>Conceptos del modelo de dominio</i>	34
2.3	REQUISITOS DEL SISTEMA	35
2.3.1.	<i>Requisitos funcionales</i>	35
2.3.2.	<i>Requisitos no funcionales</i>	36
2.4	DEFINICIÓN DE ACTORES Y DE CASOS DE USO (CU)	38
2.4.1.	<i>Listado de casos de uso del sistema</i>	38
2.4.2.	<i>Diagrama de casos de uso del sistema</i>	38
2.4.3.	<i>Descripción de casos de uso</i>	39
2.5	MODELO DE DISEÑO	42
2.5.1.	<i>Diagrama de clases del diseño</i>	42
2.5.2.	<i>Diagrama de colaboración</i>	44
2.5.3.	<i>Diagrama de paquetes</i>	48
2.6	MODELO DE DATOS FÍSICO	49
2.7	ARQUITECTURA DEL SISTEMA	51
2.8	PATRONES DE DISEÑO	52
CAPÍTULO 3.	IMPLEMENTACIÓN Y PRUEBAS	52
3.1	MODELO DE IMPLEMENTACIÓN	55
3.1.1.	<i>Estándares de codificación</i>	55
3.1.2.	<i>Diagrama de componentes</i>	56
3.1.3.	<i>Diagrama de despliegue</i>	58
3.2	PRUEBAS	59
3.3	ESTRATEGIA DE PRUEBA	60
3.3.1.	<i>Niveles de prueba</i>	60
3.3.2.	<i>Tipos de prueba</i>	61
3.3.3.	<i>Método de prueba</i>	63
3.3.4.	<i>Técnica de prueba</i>	64
3.3.5.	<i>Diseño de casos de prueba</i>	64
3.4	RESULTADO DE LAS PRUEBAS	65
CONCLUSIONES		67
RECOMENDACIONES		68
REFERENCIAS BIBLIOGRÁFICAS		69
ANEXOS		¡ERROR! MARCADOR NO DEFINIDO.

Índice de figuras

<i>Figura 1: Modelo de dominio</i>	34
<i>Figura 2: Diagrama de casos de uso del módulo Referencias Bibliográficas</i>	38
<i>Figura 3: Diagrama de clases con estereotipos web del caso de uso Gestionar lista de documentos</i>	42
<i>Figura 4: Diagrama de clases con estereotipos web del caso de uso Eliminar ítems de una lista</i>	43
<i>Figura 5: Diagrama de clases con estereotipos web del caso de uso Crear referencias bibliográficas</i>	43
<i>Figura 6: Diagrama de clases con estereotipos web del caso de uso Exportar referencias bibliográficas</i>	44
<i>Figura 7: Diagrama de clases con estereotipos web del caso de uso Enviar referencias bibliográficas por correo electrónico</i>	44
<i>Figura 8: Diagrama de colaboración del requisito funcional Eliminar ítems de una lista</i>	45
<i>Figura 9: Diagrama de colaboración del requisito funcional Crear referencias bibliográficas</i>	45
<i>Figura 10: Diagrama de colaboración del requisito funcional Exportar referencias bibliográficas</i>	46
<i>Figura 11: Diagrama de colaboración del requisito funcional Enviar referencias bibliográficas por correo electrónico</i>	46
<i>Figura 12: Diagrama de colaboración del requisito funcional Adicionar lista</i>	46
<i>Figura 13: Diagrama de colaboración del requisito funcional Editar lista</i>	47
<i>Figura 14: Diagrama de colaboración del requisito funcional Eliminar lista</i>	47
<i>Figura 15: Diagrama de colaboración del requisito funcional Visualizar lista</i>	48
<i>Figura 16: Diagrama de paquetes</i>	48
<i>Figura 17: Modelo de datos del módulo Referencias bibliográficas</i>	50
<i>Figura 18: Arquitectura de tres capas en el módulo Referencias Bibliográficas</i>	52
<i>Figura 19: Diagrama de componentes del módulo Referencias Bibliográficas</i>	57
<i>Figura 20: Diagrama de componentes del subsistema Aplicación</i>	57
<i>Figura 21: Diagrama de componentes del subsistema negocio</i>	58
<i>Figura 22: Diagrama de componente del subsistema Acceso a Datos</i>	58
<i>Figura 23: Diagrama de despliegue del módulo Referencias Bibliográficas</i>	59
<i>Figura 24: Informe Agregado de la prueba de carga al módulo Referencias Bibliográficas</i>	63
<i>Figura 25: Resultados de las pruebas de caja negra aplicadas al módulo Referencias Bibliográficas del sistema REPXOS 3.0</i>	66

Índice de tablas

<i>Tabla 1: Descripción de actores del sistema</i>	<i>38</i>
<i>Tabla 2: Descripción de caso de uso Gestionar lista de documentos registrados en el sistema</i>	<i>39</i>
<i>Tabla 3: Descripción del caso de uso Eliminar ítem de una lista</i>	<i>39</i>
<i>Tabla 4: Descripción del caso de uso Crear referencias bibliográficas</i>	<i>40</i>
<i>Tabla 5: Descripción del caso de uso Exportar referencias bibliográficas</i>	<i>40</i>
<i>Tabla 6: Descripción del caso de uso Enviar referencias bibliográficas por correo electrónico</i>	<i>41</i>
<i>Tabla 7: Prueba de caja negra del caso de uso Crear referencias bibliográficas</i>	<i>65</i>
<i>Tabla 8: Descripción de las variables del caso de uso Crear referencias bibliográficas</i>	<i>65</i>

Introducción

Desde hace varios años, los profesionales de la informática tienen que afrontar el reto de solucionar los problemas de recuperación de información, provocados por la aparición de las bibliotecas digitales y el aumento vertiginoso de la información en formato electrónico. Con el acelerado desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), se ha logrado en gran medida transformar la manera en que se preserva ese crecimiento exponencial de la información, evidenciándose en la forma de organizarla, divulgarla y administrarla eficientemente.

Para las instituciones es de vital importancia utilizar los recursos de las TIC para generar, conservar, e intercambiar gran parte de su patrimonio intelectual. Es por ello que se crean los Repositorios Digitales de Acceso Abierto (RDAA), que son un conjunto de servicios de almacenamiento, gestión y distribución de materiales digitales disponibles para los miembros de una determinada comunidad. (1)

Los Repositorios Digitales han ido ocupando rápidamente un lugar importante en las universidades e instituciones docentes y científicas de todo el mundo. La creación de los mismos goza de gran aceptación a escala mundial, fundamentalmente con la proliferación del desarrollo de herramientas que sustentan su implementación. (1)

Las fuentes de información normalmente tienden a crecer de forma descontrolada, de modo que al utilizar los métodos manuales para gestionar toda esta información, se hace muy complejo y engorroso realizar una adecuada labor en la gestión de la bibliografía. Este es uno de los motivos por lo que algunos usuarios renuncian a reconocer la importancia de los materiales que han contribuido en la realización de su trabajo, algo inaceptable desde el punto de vista ético y legal en el ámbito de la ciencia.

En el curso de un estudio o una investigación se encuentran muchas publicaciones, que contienen información relevante que posteriormente se va a necesitar. Para identificarlas fácilmente, se necesita registrar los detalles. Estos detalles son conocidos como referencias bibliográficas. Por tanto, una de las funciones o aplicaciones de la referencia bibliográfica es la de sustentar una investigación científica, histórica o social, informando al lector sobre las fuentes de citas o alusiones tomadas de otros trabajos. (2)

La etapa crucial del proceso bibliográfico es el momento en el cual se consulta una fuente y se decide tomar alguna idea, texto o cierto tipo de información de ella. Este es exactamente el momento cuando el lector debe anotar o tomar los datos requeridos según el tipo de fuente de que se trate. Haber establecido el estilo de descripción que utilizará para realizar la bibliografía de su trabajo, es una decisión previa muy importante, porque de ella dependen, en cierta medida, los datos y la forma en que los debe

tomar. Esto reviste una importancia especial en el caso de los recursos dispuestos en Internet. Puede que unos días después sea demasiado tarde para reencontrar el documento consultado, además del esfuerzo que implica reencontrarlo. (3)

Elementos como el título, el autor, la fecha de edición y otros, que describen la fuente donde se extrajo la información (libro, revista, tesis, página web y otros) se unen en la bibliografía sobre la base de un formato o estilo bibliográfico, que indican la forma y el orden en que deben presentarse estos elementos en las referencias bibliográficas (3). Para asegurar el acceso a dicha información, la misma puede gestionarse a través de un software que permita ubicarla de manera coherente dentro de la Web, facilitando su recuperación a través de herramientas de búsqueda por medio de metadatos asociados a cada objeto digital.

Un metadato describe los atributos de un recurso, teniendo en cuenta que el mismo puede consistir en un objeto bibliográfico, registros e inventarios archivísticos, objetos geoespaciales, recursos visuales y de museos o implementaciones de software. Aunque puedan presentar diferentes niveles de especificidad o estructura, el objetivo principal es el mismo: describir, identificar y definir datos. (4)

La Universidad de las Ciencias Informáticas (UCI) cuenta con varios centros de desarrollo de software entre los que se encuentra el Centro de Informatización de la Gestión Documental (CIGED). Este centro desarrolla un sistema para la creación de Repositorios Digitales nombrado REPXOS 3.0, donde se centraliza la documentación digital que contiene la producción científico-técnica e intelectual de una institución. Dicho sistema cuenta con varias funcionalidades que permiten realizar búsquedas sobre los documentos que almacena.

Una de las limitaciones que se ha identificado en el sistema REPXOS 3.0, es que no les permite a sus usuarios extraer las referencias bibliográficas de las fuentes de información consultadas, sin tener que hacer uso de gestores bibliográficos para la confección de estas. Esta situación trae consigo que el proceso se realice de forma manual, lo que puede provocar errores de redacción a la hora de introducir los datos, además de pérdida de información útil, pues muchas veces se consultan varios materiales localizados en distintos lugares y se olvida incluirlos como parte de la investigación. Todo esto hace que se convierta en un proceso lento, engorroso y complicado, mucho más cuando son varias las fuentes consultadas. Además no incluye una funcionalidad que permita el envío de las referencias bibliográficas previamente conformadas por el usuario mediante correo electrónico, limitando la satisfacción de los investigadores, sus exigencias y expectativas.

Por lo antes planteado se identifica como **problema a resolver**: ¿cómo gestionar las referencias bibliográficas de los documentos científicos consultados en el Sistema para repositorios digitales REPXOS 3.0?

Este problema se enmarca en el **objeto de estudio**: proceso de extracción y conversión de metadatos de documentos científicos en sistemas que generan referencias bibliográficas; incidiendo en el **campo de acción**: proceso de extracción y conversión de metadatos para la generación de referencias bibliográficas.

Para dar solución al problema se plantea como **objetivo general**: desarrollar un módulo que permita la gestión de las referencias bibliográficas de los documentos científicos consultados en el Sistema para repositorios digitales REPXOS 3.0, utilizando estándares y normas internacionales, a partir del proceso de extracción y conversión de metadatos.

Para dar cumplimiento al objetivo general propuesto se llevarán a cabo las siguientes **tareas de investigación**:

- Estudio de los sistemas informáticos nacionales e internacionales que implementan componentes de software de Referencias Bibliográficas para analizar sus características, ventajas y desventajas.
- Caracterización de la metodología de desarrollo de software, herramientas, tecnologías y lenguajes de desarrollo que conforman la línea base de la arquitectura para el desarrollo del módulo Referencias Bibliográficas.
- Obtención de los artefactos que se generan a partir de la metodología de desarrollo seleccionada.
- Implementación del módulo Referencias Bibliográficas para dar solución a los requisitos.
- Realización de pruebas al módulo Referencias Bibliográficas para validar su correcto funcionamiento.

El Sistema para repositorios digitales REPXOS 3.0 ofrece una amplia difusión de toda la producción intelectual digital generada en una institución. Puede incrementar la visibilidad y el prestigio de la misma, así como aumentar la visibilidad de sus investigadores. Ampliando la difusión y el uso de sus trabajos, los autores pueden ver incrementado el impacto de sus investigaciones. Cuando un usuario accede al sistema para consultar un documento científico este se encuentra en la necesidad de referenciar dicho documento en su trabajo investigativo, mediante una referencia bibliográfica.

En aras de permitirle al usuario que acceda a REPXOS 3.0 gestionar las referencias bibliográficas sin tener que utilizar métodos manuales para conformar las mismas, se crea el módulo Referencias Bibliográficas. Entre los múltiples beneficios que aporta el módulo se encuentran que permite el envío

de las referencias bibliográficas por correo electrónico, y la exportación y generación de las mismas sin tener que hacer usos de gestores bibliográficos externos. Implementa una gran gama de estilos bibliográficos y formatos de archivos, que son adaptables a cualquier institución independientemente de su campo de especialización. Para las instituciones, contar con este tipo de repositorio puede incrementar su visibilidad y prestigio. Para los investigadores, aumenta la visibilidad de sus trabajos, ampliando la difusión y el uso de los mismos.

En el curso de la investigación se utilizaron **métodos científicos** para dar cumplimiento a las tareas de investigación, los cuales se clasifican en dos grupos: teóricos y empíricos.

Teóricos:

Analítico-Sintético: para realizar un análisis de las teorías y conceptos sobre la extracción y conversión de metadatos para la generación de referencias bibliográficas, permitiendo la valoración de los elementos más importantes que definen la necesidad de la investigación.

Modelación: para representar los procesos y prototipos correspondientes a la fase de análisis y diseño, proporcionando una mejor perspectiva del módulo que se quiere desarrollar.

Empíricos:

Observación: para tener una visión clara y completa de los elementos que están presentes en el dominio de la investigación.

La presente investigación está estructurada en tres capítulos, organizándose la información contenida de la siguiente manera:

Capítulo 1: Fundamentación Teórica.

En este capítulo se ofrecen algunas consideraciones teórico-metodológicas acerca de la extracción y conversión de metadatos de documentos científicos, donde se fundamenta además la metodología, tecnologías y herramientas utilizadas para el desarrollo del módulo Referencias Bibliográficas.

Capítulo 2: Características del módulo Referencias Bibliográficas.

Se realiza una caracterización del módulo a desarrollar, así como también de las funcionalidades a implementar. Se definen los diagramas, requisitos funcionales, no funcionales, arquitectura del sistema, patrones utilizados y se realizan los resúmenes de los casos de usos.

Capítulo 3: Construcción del módulo Referencias Bibliográficas.

Se presentan, el modelo de despliegue y los componentes que conforman la solución. Se realiza la validación de la misma, exponiendo el resultado final de las pruebas realizadas. Además se muestran los estándares de codificación utilizados para la solución del módulo.

CAPÍTULO 1. Fundamentación teórica

En el presente capítulo se analizan las tendencias y tecnologías existentes en el mundo, relacionadas con la generación de referencias bibliográficas. Se definen los conceptos básicos asociados al dominio del problema, y se fundamentan las tecnologías, herramientas, lenguajes de programación y metodología de desarrollo de software a utilizar en la implementación del sistema, lo que constituye la base teórica de la investigación.

1.1 Metadatos

Los metadatos son datos altamente estructurados que describen información: el contenido, la calidad, la condición y otras características de los datos. El término metadatos describe varios atributos de los objetos de información y les otorga significado, contexto y organización. En el mundo digital, han aparecido categorías de metadatos adicionales para sustentar la navegación y la gestión de archivos. (5)

Los datos describen el mundo real y son un modelo de la realidad; los metadatos describen los datos y se utilizan para tomar decisiones acerca de los mismos. Constituyen la información, en forma de documentación que permite que los datos sean bien entendidos, compartidos y explotados de manera eficaz por todo tipo de usuarios a lo largo del tiempo. Se utilizan para poder identificar, acceder y usar los datos. (6)

Con fines prácticos, los tipos y funciones de los metadatos pueden clasificarse en tres amplias categorías: descriptivos, estructurales y administrativos. Estas categorías no siempre tienen límites bien definidos y con frecuencia presentan un significativo nivel de superposición. (5)

Metadatos Descriptivos: se utilizan para la descripción e identificación de la información contenida en el recurso. Contienen atributos físicos (medios, condición de las dimensiones) y atributos bibliográficos (título, autor/creador, idioma, palabras clave). (7)

Metadatos Administrativos: se refieren a las características y propiedades del recurso, facilitando la gestión y procesamiento tecnológico y físico de las colecciones digitales tanto a corto como a largo plazo. Incluyen información sobre la creación y el control de calidad, gestión de derechos, control de acceso y utilización y condiciones de preservación. (7)

Metadatos Estructurales: proporcionan información sobre la estructura interna de los recursos electrónicos, como página, sección, capítulo, partes, índices, tabla de contenido, etc. y describen la relación entre los materiales. Facilitan la navegación y presentación de los recursos y relacionan las diferentes partes que lo componen. (7)

En el sistema REPXOS 3.0 se utilizan los metadatos descriptivos para manejar los datos contenidos en el repositorio, pues los mismos describen e identifican los recursos de información para su posterior búsqueda y recuperación, así como la localización cuando se trata de un entorno web.

1.2 Extracción y conversión de metadatos de un documento

La extracción automática de metadatos en sistemas que manejan RDAA, consiste en obtener un conjunto de atributos o elementos necesarios que describan un documento digital, como son el título, autor, año de publicación, entre otros. Los metadatos extraídos se utilizan para conformar las referencias bibliográficas de los documentos digitales, dependiendo de una norma o estandarización.

1.2.1. Formatos de archivos más utilizados por gestores bibliográficos

Los gestores bibliográficos son aplicaciones informáticas destinadas a manejar bases de datos de referencias bibliográficas, obtenidas a partir de distintas fuentes de información y son capaces de crear, mantener, organizar y dar forma a dichas referencias según diferentes normas bibliográficas. (8)

Los gestores más utilizados por su calidad, prestaciones y por las mejoras introducidas en los últimos años son Mendeley, RefWorks, EndNote Web y Zotero (9). Estos poseen funciones similares aunque su nivel de prestaciones es diferente, por lo que son analizados para conocer cuáles son los formatos de archivos que permite importar o exportar, y así poder usar este conocimiento como base de la solución.

Actualmente, existe una gran variedad de estos programas. A continuación se describen los más utilizados mundialmente por investigadores, para gestionar sus bibliografías, así como los principales formatos de archivos que permiten importar:

Mendeley: es un sistema de gestión de referencias bibliográficas, que permite organizar tanto la información bibliográfica, como adjuntar los documentos a texto completo. Funciona como un programa que se descarga y se ejecuta en ordenadores personales o en dispositivos móviles, pero también como un servicio en línea, a través del cual se puede organizar y compartir la información bibliográfica con otros usuarios. (10)

- Exporta e importa ficheros bibliográficos de otro software como Zotero, RefWorks y EndNote. Asimismo exporta e importa al formato RIS y BibTeX. (11)

Zotero: es un programa de software libre para la gestión de referencias bibliográficas. Es una extensión libre para el navegador Firefox, que permite a los usuarios recolectar, administrar y citar investigaciones de todo tipo. Importa datos directamente desde las páginas web visualizadas en el momento. (10)

- Exporta ficheros hacia BibTeX, EndNote, Refer, BibIX, MODS XML y RIS.
- Importa ficheros desde BibTeX, EndNote/Refer/BibIX, ISI (parcialmente), MODS XML, Ovid (parcialmente), *PubMed*, RIS (parcialmente).

- Incorpora estilos de citas como: APA, Chicago/Turabian, Harvard, MLA, entre otros.(2)

EndNote Web: es un gestor de referencias integrado en la plataforma *ISI Web of Knowledge*. Está diseñado para ayudar a los estudiantes e investigadores en el proceso de escritura de un trabajo de investigación. Permite recuperar rápida y fácilmente referencias desde una amplia variedad de fuentes de datos mediante la exportación directa, la búsqueda en línea o la importación de archivos de texto. (12)

- Incorpora estilos bibliográficos muy utilizados en el ámbito académico y de la investigación como: APA, Chicago, Harvard, MLA.
- Permite importar ficheros: BibTeX (parcialmente), CSA, EndNote/Refer/BibIX, ISI, Medline, Ovid, PubMed, RIS, SciFinder.
- Posee formatos de archivo de listas de referencias: HTML, RTF, TXT y otros como: *clipboard* y XML. (3)

RefWorks: es una herramienta en línea de gestión de bibliografía. Con ella se pueden importar, organizar, exportar y compartir referencias bibliográficas, así como darles el formato adecuado. (13)

- RefWorks permite añadir referencias de bases de datos automáticamente de forma directa, o de forma manual, y requiere guardar los registros en un documento con extensión TXT, e importar los registros del documento guardado. Además permite importar archivos en formato RIS.(14)

Los formatos de archivo utilizados en la investigación para exportar las referencias bibliográficas fueron seleccionados por su compatibilidad con los gestores bibliográficos más utilizados actualmente por investigadores en todo el mundo, para poder ser importados en los mismos. Estos formatos seleccionados son: **XML, BibTeX, RIS, RTF, TXT y HTML**. La mayoría de los gestores bibliográficos importan los formatos mencionados anteriormente, lo que permite darle una mayor importancia al módulo a desarrollar.

1.3 Estilos bibliográficos

Los estilos bibliográficos juegan un papel fundamental a la hora de conformar las referencias bibliográficas de una investigación, pues no basta solo con disponer de los datos de los materiales consultados, también se debe seguir una norma que permita organizar de forma coherente y uniforme todos esos datos. Se considera un estilo bibliográfico al conjunto de normas que indican la información que se ha de incluir, así como el orden en que deben presentarse al citar, cuando se elabora un texto y hacer una bibliografía de acuerdo a la fuente de información (15).

El estilo bibliográfico debe elegirse antes de iniciar una investigación, escrito o publicación y debería hacerse de acuerdo con el tipo de escrito y su campo de especialización. No conviene, por ejemplo, seguir un estilo diseñado para el área de humanidades en un campo científico. Cada disciplina tiene sus

particularidades y convenciones; por lo tanto, siempre conviene informarse del campo propio, los estilos más difundidos entre la comunidad de científicos para la que se escribe y elegir el sistema más versátil para las particularidades bibliográficas de la investigación. (16)

1.3.1. Estilos bibliográficos más utilizados para conformar las referencias bibliográficas

Cuando se va a citar y presentar la lista de referencias bibliográficas, se pueden emplear diferentes estilos bibliográficos y normas internacionales de redacción de bibliografías. La selección del estilo o norma puede venir recomendada por el propio profesor al que se debe presentar el trabajo; prescrita por la propia institución o universidad donde se estudia, o incluso por los propios departamentos; por la revista o congreso, en el caso de publicaciones de investigadores; o por organismos que disponen de su propio estilo. Entre los estilos bibliográficos más difundidos y utilizados se encuentran los que a continuación se presentan:

APA: es el estilo de citas bibliográficas oficial de la Asociación Psicológica de América (APA por sus siglas en inglés) y establece el formato para todo tipo de citas y documentos en Psicología y Ciencias Sociales. Fue creado en 1929 y contempla la estructura de los documentos, longitud, puntuación, abreviaturas, cuadros, partes del manuscrito y citas bibliográficas. (17)

IEEE: el estilo IEEE se utiliza principalmente en las publicaciones técnicas de ingenierías y ciencias informáticas. Una característica que lo diferencia de otros estilos es la de utilizar corchetes, en vez de superíndices, para numerar las citas tanto en contexto como en la lista de referencias. Además esta lista de referencias u obras citadas no va por orden alfabético sino numérico. (17)

Vancouver: creado en 1978, es el estilo establecido por el Comité Internacional de Editores de Revistas Médicas (ICMJE por sus siglas en inglés). Es conocido igualmente como Requisitos de Uniformidad para Manuscritos Enviados a Revistas Biomédicas (URM por sus siglas en inglés). Es una norma de la *American National Standards Institute* (ANSI) y como tal es usado por las principales revistas de Medicina, además de la *National Library of Medicine* (NLM) desde 1979 y por PubMed. El objetivo es tanto científico como ético. URM procura establecer un modo claro, sencillo y pertinente para la distribución de estudios e informes biomédicos. El estilo Vancouver establece normas y recomendaciones para la publicación de artículos científicos, citas bibliográficas, ediciones, revisiones y para los títulos abreviados de las revistas, que deben ser los del *Index Medicus*. (17)

Cada referencia utilizada tiene un número y las citas tienen que ir numeradas, el número es obligatorio en el contexto de la cita. Los títulos de publicaciones periódicas deben citarse en su forma abreviada, siguiendo las abreviaturas del *PubMed Journal Database*. (17)

Harvard: más que un estilo bibliográfico, Harvard es una forma de realizar las citas en contexto y de hacer las correspondientes listas de referencias. Tiene su origen en 1881 en la Universidad de Harvard. Su principal característica es la utilización de las citas en contexto con el formato autor-fecha: (autor, fecha) o autor (fecha), dependiendo de los casos. La lista de referencias se hace al final del documento, colocando los autores por orden alfabético y en caso de que un autor tenga diferentes citas, sus referencias van por orden cronológico. Esta forma de hacer las citas se aplicó a la mayoría de los posteriores estilos bibliográficos como MLA y APA. (17)

ISO-690: es una norma de la Organización Internacional de Normalización (ISO por sus siglas en inglés) que especifica los elementos que deben ser incluidos en las referencias bibliográficas de materiales publicados, como monografías y publicaciones seriadas, capítulos, artículos de publicaciones seriadas (como revistas y diarios), materiales cartográficos, grabaciones sonoras, fotografías, obras audiovisuales y documentos de patentes. (17)

Da un orden prescrito a los elementos de la referencia y establece convenciones para la transcripción y presentación de información derivada de la publicación fuente, aunque las cuestiones de estilo (puntuación, estilos tipográficos, mayúsculas) no son, en general, parte de la norma. ISO 690 cubre referencias a materiales publicados en forma impresa o no impresa. Sin embargo, no se aplica a referencias de manuscritos u otros materiales no publicados. (17)

Chicago (CMS): uno de los estilos bibliográficos más antiguos y más conocidos. El CMS es toda una normativa de estilo de edición, las recomendaciones tratan de puntuaciones, organización del texto, pies de página, notas y citas. El manual se publicó por primera vez en 1903 con la intención de unificar los criterios de citas bibliográficas de los investigadores. Usado principalmente en las materias de historia, ciencias sociales, arte, musicología y literatura. (17)

Turabian: es un estilo creado por Kate Turabian en 1937, muy similar al Chicago, aunque más sencillo y pensado para estudiantes y doctorandos. Es usado principalmente en disciplinas de Humanidades. Turabian contempla las citas a pie de página o citas al final de la parte, y bibliografías. No obstante, las citas en contexto también se especifican. (17)

MLA: es el estilo de la Asociación de Lenguas Modernas (MLA por sus siglas en inglés), utilizado para las humanidades. MLA indica no solo el estilo para hacer las citas sino también el estilo para escribir: tipografía, tamaño, calidad de papel, formato de párrafos, de enlaces y puntuación, especialmente para los escritos de lenguas modernas, crítica literaria y escritos culturales. Es más exhaustivo que los estilos de citas Chicago y Turabian. MLA da prioridad a las citas entre paréntesis, y con ello establece las

normas para las citas a pie de página o las citas a final de capítulo. La lista de obras citadas es obligatoria, MLA contempla asimismo formato de bibliografías. (17)

1.4 Análisis de sistemas que implementan componentes de software de Referencias Bibliográficas

Se realizó un análisis a este tipo de sistemas y se llegó a la conclusión que los que más se asemejan a la solución del módulo Referencias Bibliográficas son los siguientes:

1.4.1. Repositorio institucional de la Universidad Oberta de Catalunya (UOC), España

El Repositorio Institucional de la UOC es el portal para la recogida, difusión y preservación de las publicaciones digitales con acceso abierto a sus miembros, estas son elaboradas en el desarrollo de sus actividades de investigación, docencia y gestión. La comunidad de docencia contiene los materiales docentes y los objetos de aprendizaje generados en la actividad docencia-aprendizaje tanto por el profesorado de la UOC (recursos de aprendizaje: manuales, guías de estudio, ejercicios, pruebas de evaluación continuada, exámenes, revistas, etc.) como por alumnos (trabajos finales de grado y de máster). El proyecto de implantación del repositorio institucional de la UOC, surgió de la necesidad de aprovechar el conocimiento que se genera en la universidad por todos sus miembros, facilitando un acceso centralizado, de forma fácil, tanto por sus miembros como por la sociedad en general. (18)

Entre las características favorables que presenta el sistema se tiene, que está basado en la herramienta de creación de repositorios Dspace, lo que contribuye al aprovechamiento de la forma de almacenar y extraer los metadatos de los documentos consultados. Además permite exportar las referencias bibliográficas al gestor de contenidos RefWorks. Este sistema también presenta limitantes como por ejemplo, que es un software privativo por lo que se tiene acceso limitado a su código fuente, además de que no se pueden exportar las referencias bibliográficas a un formato de archivo determinado, ni tampoco permite enviarlas por correo electrónico a un usuario determinado.

1.4.2. Biblioteca Campus Cuernavaca de la Universidad Tecnológica de Monterrey, México

El Tecnológico de Monterrey es un sistema universitario que tiene como misión formar personas íntegras, éticas, con una visión humanística y competitiva internacionalmente en su campo profesional, que al mismo tiempo sean ciudadanos comprometidos con el desarrollo económico, político, social y cultural de su comunidad y con el uso sostenible de los recursos naturales. (20)

Cuenta con una biblioteca virtual que tiene como misión apoyar la labor académica y de investigación de la comunidad del campus, a través de recursos y servicios de información acordes con

los programas de estudio vigentes. La biblioteca presenta una funcionalidad que permite armar una ficha bibliográfica en el estilo APA, con la finalidad de apoyar en la generación de bibliografía para los trabajos académicos y de investigación. A pesar de ello, no permite exportar dicha referencia a ningún gestor bibliográfico. Este sistema es un software privativo por lo que se tiene acceso limitado a su código fuente, además presenta una gama reducida de estilos bibliográficos para la conformación de las referencias bibliográficas. Tampoco permite enviar por correo electrónico las referencias bibliográficas de los documentos consultados.

1.4.3. Google Académico

Es un buscador de Google que permite realizar búsquedas especializadas de manera sencilla, de materiales más relevantes dentro del mundo de la investigación académica. Las búsquedas las realiza en un gran número de disciplinas y fuentes como por ejemplo, estudios revisados por especialistas, tesis, libros, resúmenes y artículos de fuentes, como editoriales académicas, sociedades profesionales, depósitos de impresiones preliminares, universidades y otras organizaciones académicas. Cuenta con una funcionalidad llamada Citar que aparece en la parte inferior de un documento consultado, que permite copiar y pegar la referencia de dicho documento en un formato específico o importarla a un gestor bibliográfico. Además cuenta con una funcionalidad que permite importar directamente cuando se realiza una consulta, a algún gestor bibliográfico previamente seleccionado. (21)

Puesto que la mayor parte de resultados de búsqueda de Google Académico enlazan directamente a artículos de publicaciones comerciales, la mayor parte del tiempo los usuarios solo podrán acceder a un breve resumen del tema tratado en el artículo, así como a pequeños fragmentos de información importante relacionada con dicho artículo, y posiblemente tengan que pagar un determinado importe para acceder al artículo completo. Es un software privativo por lo que se tiene acceso limitado a su código fuente. Este sistema no permite enviar las referencias bibliográficas conformadas por correo electrónico a un usuario determinado.

1.4.4. Sistema Integrado de Gestión Bibliotecaria (SIGB) de la UCI, Cuba

Este tipo de sistemas son aplicaciones informáticas creadas con el objetivo de automatizar los sistemas y entornos bibliotecarios, aplicables a servicios propios de distintos tipos de bibliotecas ya sean públicas o privadas. Los recursos humanos de la biblioteca utilizan dispositivos y programas informáticos, adecuados a la naturaleza de los datos, para realizar procesos y facilitar los servicios que permiten alcanzar el objetivo de la biblioteca: almacenar de forma organizada el conocimiento humano contenido en todo tipo de materiales bibliográficos para satisfacer las necesidades informativas, formativas, recreativas y/o de investigación de los usuarios. (19)

Entre las características favorables que presenta este sistema se encuentra, que permite exportar las referencias bibliográficas a varios formatos de archivos de los más utilizados mundialmente, además estas referencias pueden ser enviadas por correo electrónico a un usuario determinado. Una de sus limitantes es que tiene una gama reducida de estilos bibliográficos para la conformación de las referencias bibliográficas.

1.5 Conclusiones del análisis de sistemas similares

El estudio de los sistemas anteriormente descritos, arrojó como resultado que 3 de ellos están creados sobre tecnologías privativas, por lo que no pueden utilizarse en el desarrollo del módulo Referencias Bibliográficas para el sistema REPXOS 3.0, debido a los costos por conceptos de licencias y a las políticas existentes en Cuba y la UCI de promover el desarrollo de software utilizando tecnologías de código abierto, con el objetivo de alcanzar la soberanía tecnológica del país. Se analizaron las ventajas que garantizaban el uso de estos sistemas, las cuales sirvieron como base para la implementación del módulo Referencias Bibliográficas. Además, se tuvieron en cuenta las limitantes presentadas por los mismos, siendo objeto de solución en el desarrollo del módulo propuesto.

1.6 Tecnologías y herramientas utilizadas en el desarrollo

Seleccionar correctamente las tecnologías y herramientas a utilizar en la creación de un software, posibilita ahorrar tiempo y en gran medida la calidad del producto final.

El módulo Referencias Bibliográficas se acoge a la arquitectura del sistema REPXOS 3.0, por lo que para su implementación se utilizarán las técnicas, herramientas, metodología y flujos de procesos definidos para desarrollar este sistema.

1.6.1. Metodología de desarrollo de software

Rational Unified Process (RUP): Metodología de desarrollo de software que provee una orientación disciplinada para asignar tareas y responsabilidades dentro de una organización que produce software. Su meta es asegurar la realización del software con la calidad requerida para resolver las necesidades de los usuarios finales y clientes dentro de un presupuesto y tiempo establecidos. (22)

RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). (23)

El sistema para repositorios digitales REPXOS 3.0 se desarrolló siguiendo esta metodología, por lo tanto en el desarrollo del módulo Referencias Bibliográficas se generan y consumen artefactos basados en ella.

1.6.2. Plataforma de desarrollo Java Development Kit 7.0 (JDK)

JDK es el kit de desarrollo oficial del lenguaje de programación Java, uno de los más populares entre los orientados a objetos. Además de la máquina virtual de Java, indispensable para ejecutar las clases de los programas, JDK cuenta con un amplio surtido de herramientas, como “javac” el compilador de *bytecode* de Java, el desensamblador de clases, y “jdb”, el depurador de bugs. (24)

A continuación se muestran sus principales características:

- Es el kit básico para programar en Java
- Instalación rápida
- Abundante documentación en línea

JDK es la plataforma sobre la cual se va a desarrollar y ejecutar el módulo Referencias Bibliográficas, además se utilizan sus herramientas para la compilación del mismo.

1.6.3. Lenguajes de programación

Java: lenguaje orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. (25)

Provee numerosas comprobaciones en compilación y en tiempo de ejecución, está creado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. (25)

Java es el lenguaje de programación del lado del servidor que se va a utilizar para el desarrollo del sistema REPXOS y por consecuente del módulo Referencias Bibliográficas. Su uso permitió la programación de las clases asociadas a la capa lógica de negocio y almacenamiento.

JavaScript: es el lenguaje interpretado más utilizado, principalmente en la construcción de páginas web, con una sintaxis muy semejante a Java y a C. Al contrario que Java, no se trata de un lenguaje orientado a objetos propiamente dicho, sino que éste está basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. (26)

JavaScript es el lenguaje de programación del lado del cliente, utilizado para la manipulación de datos en las interfaces de usuario del módulo Referencias Bibliográficas.

Librería JavaScript

Jquery 1.10.2: librería JavaScript que simplifica tareas comunes y proporciona una capa de compatibilidad entre navegadores. Ofrece una infraestructura con la que se tiene mayor facilidad para la creación de aplicaciones complejas del lado del cliente. (26)

Esta librería de JavaScript es usada para la validación de los datos introducidos por los usuarios en la interfaces del módulo, y para la creación de los mensajes de alertas.

1.6.4. Lenguaje Unificado de Modelado (UML)

UML es un lenguaje gráfico para especificar, construir y documentar los artefactos que modelan un sistema. UML fue diseñado para ser un lenguaje de modelado de propósito general, por lo que puede utilizarse para especificar la mayoría de los sistemas basados en objetos o en componentes, y para modelar aplicaciones de muy diversos dominios de aplicación (telecomunicaciones, comercio, sanidad, etc.) y plataformas de objetos distribuidos. (27)

UML ofrece diversos elementos de esquematización que representan las diferentes partes de un sistema, de los cuales se utilizan en el desarrollo del módulo Referencias Bibliográficas los diagramas de casos de uso, de clases del diseño, de despliegue, de componentes entre otros.

1.6.5. Lenguaje para maquetado CSS 3

Permite controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. El lenguaje de hojas de estilo en cascada, es la mejor forma de separar los contenidos y su presentación es imprescindible para la creación de páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "*documentos semánticos*"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. (28)

CCS 3 es usado para el diseño y maquetación de las interfaces de usuario del módulo Referencias Bibliográficas siguiendo las pautas definidas en la estrategia marcaría de la UCI para productos de la línea XABAL.

1.6.6. Herramienta CASE

Visual Paradigm 8.0: es una herramienta CASE (Ingeniería de Software Asistida por Computación). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Esta herramienta permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería de Software. (29)

Se utiliza en el módulo Referencias Bibliográficas para la creación de diagramas requeridos por la metodología de desarrollo para guiar la implementación utilizando el Lenguaje Unificado de Modelado. Esta herramienta está disponible para múltiples plataformas lo que posibilita que se pueda trabajar sobre ella en cualquier sistema operativo, además está distribuido bajo una licencia gratuita y comercial por lo que no constituye un problema de costos. Provee el uso de un lenguaje común para todo el equipo de desarrollo.

1.6.7. Herramienta de repositorio: DSpace 4.2

DSpace es un software de código abierto diseñado por el Instituto tecnológico de Massachusetts (MIT por sus siglas en inglés) y los laboratorios de Hewlett-Packard (HP por sus siglas en inglés) para gestionar repositorios de ficheros (textuales, audio, vídeo, etc.), facilitando su depósito, organizándolos en comunidades, asignándoles metadatos y permitiendo su difusión a recolectores o agregadores. (30)

Dspace se utiliza como sistema base para el desarrollo de REPXOS 3.0, por lo que el módulo fue desarrollado siguiendo la arquitectura de Dspace, además ofrece un conjunto de metadatos registrado en el estándar *dublin-core* para la creación de las referencias bibliográficas. Dspace es además una plataforma de código abierto.

1.6.8. Herramienta de compilación: Maven 3

Es una herramienta de software para la gestión y construcción de proyectos Java. Basado en el concepto de un modelo de objetos del proyecto (POM), Maven puede gestionar de un proyecto de construcción, generación de informes y documentación de un lugar central. (31)

Maven es usada para en el proceso de compilación y construcción del sistema REPXOS, además contiene un conjunto de librerías que facilitan el mismo.

1.6.9. Servidor Apache Tomcat 7.0

Apache Tomcat es el servidor de aplicaciones determinado para el desarrollo del sistema para repositorios institucionales REPXOS 3.0 y por consecuente para el desarrollo del módulo Referencias Bibliográficas. Es una implementación de software de código abierto de Java Servlets y tecnologías Java Server Pages (JSP), es multiplataforma, funciona como contenedor de servlets y que se desarrolla bajo el proyecto denominado Jakarta perteneciente a la Apache Software Foundation bajo la licencia Apache 2.0. (33)

1.6.10. Herramienta de construcción de proyectos: Apache Ant 1.8.2

Apache Ant es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción. Esta herramienta, hecha en el lenguaje de programación Java, tiene la ventaja de no depender de las órdenes del *shell* de cada sistema operativo, sino que se basa en archivos de configuración XML y clases Java para la realización de las distintas tareas, siendo idónea como solución multiplataforma. (33)

A continuación se muestran sus principales características:

- Ant está escrito en XML y Java, esto permite ofrecer una solución interoperable al nivel de sistema operativo (debido a Java) y configuraciones descriptivas (debido a XML)
- Es multiplataforma
- Es una herramienta gratuita
- Es portable

Es utilizado en la instalación del sistema REPXOS 3.0. Ant copia el código fuente hacia la ubicación donde va a ser desplegado por el Apache Tomcat, y genera la estructura de la base de datos de REPXOS 3.0.

1.6.11. Marco de trabajo

Un marco de trabajo es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, se puede considerar como una aplicación genérica incompleta y configurable a la que se puede añadir las últimas piezas para construir una aplicación concreta. (34)

Bootstrap v2.3.2

Bootstrap es un marco de trabajo CSS, liberado por *Twitter* y muy popular hoy en día. Fue publicado en 2011 con licencia Apache. Está basado en los últimos estándares de desarrollo Web además de ser compatible con todos los navegadores habituales. (35)

La interfaz del módulo Referencias Bibliográficas está completamente diseñada con este marco de trabajo, ya que ofrece ventajas que pudieron ser aprovechadas correctamente tales como:

- Herramienta sencilla y ágil para construir sitios web e interfaces
- Utiliza componentes y servicios creados por la comunidad web, tales como: HTML5 shim, Normalize.css, OOCSS, jQuery UI, LESS y GitHub

1.6.12. Sistema Gestor de Base de Datos PostgreSQL 9.3

PostgreSQL 9.3

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. (37)

PostgreSQL está definido como el sistema gestor de base de datos a utilizar para la elaboración de las tablas necesarias para el desarrollo del módulo Referencias Bibliográficas del sistema para repositorios digitales REPXOS 3.0 ya que posee características como:

- Robustez
- Optimizador de consultas.
- Multiplataforma (Linux, Windows, Mac OS X, Solaris, y otros más).
- Ofrece disponibilidad, fiabilidad y seguridad.

1.6.13. Entorno de Desarrollo Integrado (IDE)

NetBeans 8.0: es un entorno de desarrollo integrado libre y gratuito sin restricciones de uso, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. *Sun Micro Systems* fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos. (38)

Se utiliza en el módulo Referencias Bibliográficas debido a que dispone de soporte para crear interfaces gráficas de forma visual, está diseñado íntegramente para desarrollar aplicaciones web con el lenguaje

Java que cuenta con un excelente editor de texto que facilita la creación del código fuente del sistema, se integra fácilmente a sistemas de control de versiones y además es un software multiplataforma.

Conclusiones del capítulo

Con la realización de este capítulo, se brindan elementos para lograr un mayor entendimiento de los procesos relacionados con la extracción y conversión de metadatos para la generación de las referencias bibliográficas, haciendo uso de los principales estilos bibliográficos. Con el análisis de los gestores bibliográficos más utilizados en el mundo, se pudieron identificar los formatos de archivos que estos usan para la generación de las referencias bibliográficas. Con el estudio del estado del arte se logró caracterizar sistemas similares al módulo Referencias Bibliográficas, teniendo en cuenta ventajas y limitantes, permitiendo así seleccionar las características que más se adecuan a las condiciones actuales del sistema REPXOS 3.0. Se definieron las metodologías, lenguajes y herramientas necesarias para la construcción del sistema, utilizando las que se adaptan al sistema REPXOS 3.0 y a las necesidades reales del módulo en cuestión.

CAPÍTULO 2. Características del módulo Referencias Bibliográficas

En el presente capítulo se describe la propuesta de solución para el sistema a desarrollar, sus características y funcionalidades. Se incluyen los modelos, diagramas, requisitos funcionales y no funcionales que requiere el módulo a implementar, así como los actores y descripciones de los casos de uso del sistema. Además se describe la arquitectura utilizada y los patrones de diseño.

2.1 Descripción de la propuesta de solución

Con el objetivo de facilitar el trabajo de investigadores, que consultan documentos del Repositorio Institucional de la entidad que adquiera el producto, a la hora de conformar las referencias bibliográficas, se requiere la implementación del módulo Referencias Bibliográficas que debe desarrollarse sobre la plataforma web. Su implementación proporcionará al sistema REPXOS 3.0, valiosas funcionalidades capaces de suplir las necesidades actuales del mismo.

El módulo permite a los usuarios confeccionar las referencias bibliográficas de los documentos científicos consultados que consideren útiles para su investigación, que se encuentran en las listas privadas y públicas que lo mismos manejan. El sistema es el encargado de la extracción y conversión de los metadatos de los documentos, utilizando para ello diferentes estilos bibliográficos. Los estilos que los usuarios pueden seleccionar para conformar las referencias bibliográficas son los siguientes: **APA, MLA, ISO-690, Harvard, Turabian, Vancouver, IEEE y Chicago**. Una vez conformada la referencia bibliográfica, esta se exporta a un formato de archivo (**TXT, XML, RIS, BIBTEX, HTML y RTF**) escogido por el usuario. Este archivo permite a los usuarios exportar las referencias bibliográficas creadas, a gestores bibliográficos tales como Zotero, EndNote, RefWorks, Mendeley, etc. La aplicación además permite el envío mediante correo electrónico, de las referencias bibliográficas previamente conformadas a un destinatario determinado por el usuario, adjuntándolas en un fichero con un formato de archivo previamente definido.

2.2 Modelo de dominio

El modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los conceptos que existen o los eventos que suceden en el entorno en el que trabaja el sistema. (29)

El modelo de dominio puede ser tomado como punto inicial para el diseño del sistema, este puede utilizarse para capturar y expresar los aspectos más importantes del contexto del sistema. Además es

una representación visual de los conceptos u objetos del mundo real, significativos para un problema o área de interés. A continuación se muestra el modelo de dominio de la presente investigación:

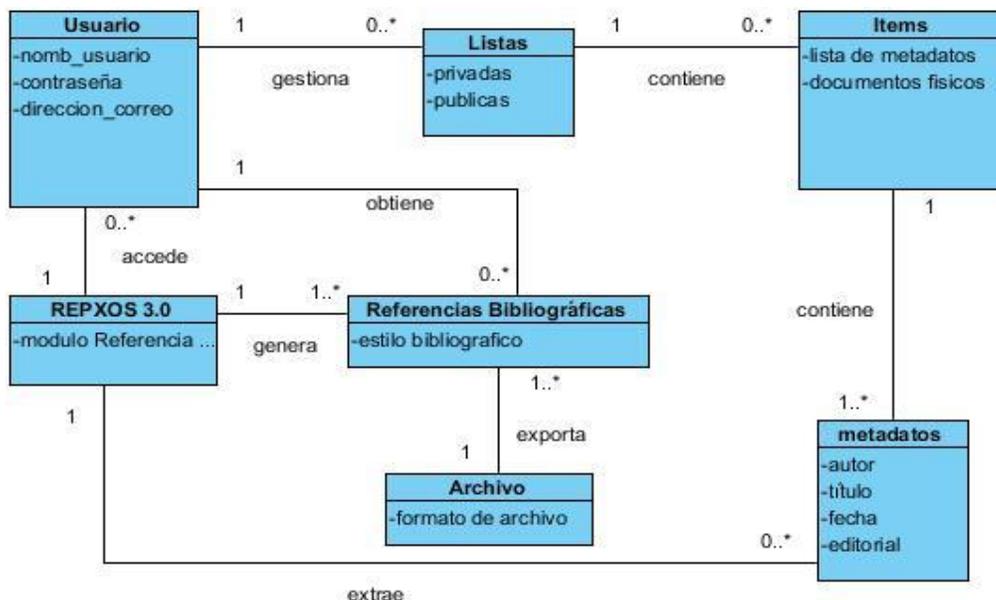


Figura 1: Modelo de dominio

2.2.1. Conceptos del modelo de dominio

Archivo: este archivo contiene un listado de las referencias bibliográficas y se exportan a un formato determinado por el usuario.

Ítems: son archivos almacenados en el sistema que representan un documento científico. Contienen una lista de metadatos.

Listas: contienen los documentos que el usuario añade. Pueden ser públicas (todos los usuarios las pueden visualizar) y privadas (solo el usuario autenticado puede ver sus propias listas).

Metadatos: son los datos de los ítems. Se describe de cada ítem el título, el autor, la fecha de publicación, etc.

Referencia Bibliográfica: es conformada con los metadatos extraídos de un ítem, seleccionando un estilo bibliográfico. Estas referencias pueden ser mostradas o enviadas al correo electrónico de un destinatario seleccionado por el usuario.

REPXOS 3.0: es el sistema encargado de automatizar los procesos y servicios que brinda el repositorio institucional a los usuarios.

Usuario: es la persona que puede gestionar las listas de documentos y puede exportar y obtener las referencias bibliográficas.

2.3 Requisitos del sistema

Los requerimientos de software son capacidades o condiciones que tienen que ser alcanzadas o poseídas por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Define las funciones de qué es lo que el sistema será capaz de realizar, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen. Son el conjunto de propiedades que debe cumplir el software para ser exitoso en el entorno en el cual se usará. Estos deben ser comprensibles por clientes, usuarios y desarrolladores, deben tener una sola interpretación y estar definidos en forma medible y verificable. (29)

2.3.1. Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones con las cuales el sistema debe cumplir e indican su comportamiento, es decir, son las funciones que el sistema debe realizar. (29)A continuación se listan y describen los requisitos funcionales que debe cumplir el sistema para la solución planteada:

RF1. Crear lista de documentos registrados en el sistema

Descripción: permite al usuario crear una lista con los documentos registrados en el sistema, que él decide referenciar. Estos van a ser guardados en la base de datos.

RF2. Eliminar Lista de documentos

Descripción: permite al usuario eliminar una lista de documentos registrados en el sistema conformada previamente.

RF3. Editar lista de documentos

Descripción: permite al usuario editar la lista de los documentos registrados en el sistema.

RF 4. Visualizar lista de documentos

Descripción: permite al usuario visualizar las listan públicas y privadas de los documentos registrados en el sistema.

RF 5. Eliminar ítem de una lista

Descripción: permite al usuario eliminar de una lista de documentos uno o varios elementos seleccionados.

RF 6. Crear Referencias Bibliográficas

Descripción: permite crear la referencia bibliográfica, a partir de la lista de documentos creada, seleccionando un estilo de referencia que pueden ser APA, ISO-690, MLA, Chicago, Turabian, IEEE, Vancouver y Harvard.

RF 7. Exportar referencias bibliográficas de ítems

Descripción: el usuario tiene la opción de exportar los registros bibliográficos de las referencias bibliográficas conformadas a un archivo en los formatos TXT, HTML, RIS, BIBTEX, XML y RTF pinchando en el botón Exportar. Estos archivos pueden ser creados seleccionando los documentos seleccionados o toda la lista de documentos.

RF 8. Enviar referencias bibliográficas por correo electrónico

Descripción: una vez que se muestre la lista de documentos escogidos por el usuario, permite enviar las referencias bibliográficas de uno o varios documentos seleccionados por el usuario, al correo electrónico de un destinatario.

2.3.2. Requisitos no funcionales

Los requisitos no funcionales son los requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste, como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. Los requisitos no funcionales representan aquellos atributos que debe exhibir el sistema pero que no son funcionalidades específicas. (29)

Los requerimientos no funcionales que debe tener el módulo son (39):

RnF 1. Usabilidad: El sistema proporcionará una interfaz sencilla y atractiva, con facilidad de uso para usuarios con conocimientos mínimos de informática. Los grupos de botones y vínculos deben estar organizados por la funcionalidad, con el objetivo de facilitar al usuario la interacción con el sistema. Los mensajes para interactuar con los usuarios y los de error deben ser lo suficientemente informativos e intuitivos, en idioma español y no deben revelar información interna. Los colores a utilizar en el desarrollo del módulo Referencias bibliográficas deben ajustarse a los colores del REPXOS 3.0.

RnF 2. Capacidad: el sistema debe ser capaz de soportar 200 usuarios conectados concurrentemente.

RnF 3. Recursos: para el correcto funcionamiento del sistema se deben tener en cuenta que el entorno de desarrollo y prueba tiene las siguientes características:

- Procesador: Intel Core i3-2120 a 3.30 Ghz
- Memoria RAM: 2 GB
- Disco Duro: 3 GB libre para instalación del sistema. Se necesita espacio adicional para el almacenamiento de los documentos digitales en dependencia de la institución que vaya a utilizar el sistema.

Por lo tanto los recursos mínimos para que el módulo funcione de forma adecuada deben estar en correspondencia con los descritos anteriormente, de lo contrario no se garantiza el correcto funcionamiento del mismo.

RnF 4. Documentación: se documenta el módulo con un manual de usuario con el objetivo de proporcionar al usuario la información necesaria para utilizar el mismo, así como detalles de su funcionamiento.

RnF 5. Lenguaje de programación: se utiliza para la construcción del sistema los lenguajes de programación Java, HTML5, JavaScript 1.10.2 y las herramientas que se utilizan son de distribución bajo licencias libres, para favorecer el desarrollo de nuevos componentes de software o subsistemas.

RnF 6. Forma de acceso: al sistema se puede acceder desde cualquier navegador web.

RnF 7. Interfaces de usuario: el módulo está optimizado para la correcta visualización en navegadores web tales como Mozilla Firefox, Google Chrome, Internet Explorer, Safari y Opera. Las interfaces web del módulo Referencias Bibliográficas fueron probadas en las versiones siguientes de los navegadores antes mencionados:

- Mozilla Firefox: versión 20 - 38.
- Google Chrome: versión 10 - 43.
- Internet Explorer: versión 6 - 11.
- Safari: versión 5 - 8.
- Opera: versión 10 - 29.

No se garantiza la correcta visualización en versiones por debajo del número mínimo de las antes mencionadas de los navegadores web.

RnF 8. Interfaces de Hardware: el hardware donde se instalará el sistema debe poseer al menos una interfaz de red cuya velocidad de transferencia iguale o supere los 100 Mbps.

RnF 9. Interfaces de Software: El sistema debe integrarse con el Gestor de base de datos PostgreSQL desde 8.4 hasta 9.3, el contenedor de Servlets utilizado como servidor web Apache Tomcat 7.0 y la herramienta de compilación Maven 3.0.

2.4 Definición de actores y de casos de uso (CU)

Los actores del sistema no son más que sujetos, objetos o un mismo sistema que interactúa con este de forma externa (29), el actor del sistema en cuestión es el siguiente:

Tabla 1: Descripción de actores del sistema

Actor	Descripción
Usuario	Es la persona que puede obtener las referencias bibliográficas seleccionando un estilo bibliográfico determinado y gestionar las listas de documentos del sistema.

2.4.1. Listado de casos de uso del sistema

- CU1-Gestionar lista de documentos registrados en el sistema
- CU2-Eliminar ítem de una lista
- CU3-Crear referencia bibliográfica
- CU4-Exportar referencias bibliográficas de ítems
- CU5-Enviar referencias bibliográficas por correo electrónico

2.4.2. Diagrama de casos de uso del sistema

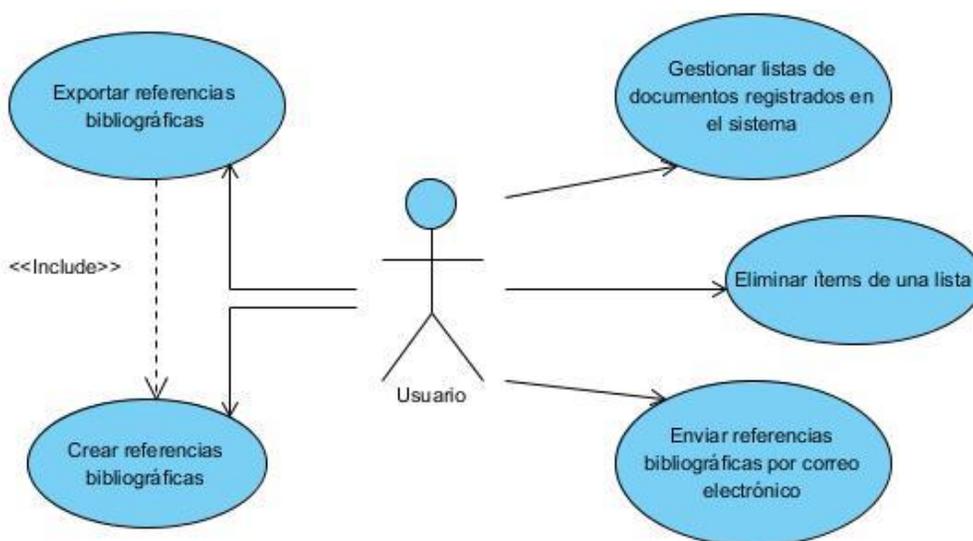


Figura 2: Diagrama de casos de uso del módulo Referencias Bibliográficas

2.4.3. Descripción de casos de uso

Tabla 2: Descripción de caso de uso Gestionar lista de documentos registrados en el sistema

Objetivo	Gestionar la lista de documentos registrados en el sistema.
Actores	Usuario (inicia): puede crear, eliminar, modificar y visualizar la lista de documentos registrados en el sistema.
Resumen	El caso de uso inicia cuando el usuario selecciona en el menú la opción Mis listas, el sistema visualiza las listas (privadas y públicas) y permite las acciones de crear, editar y eliminar sus listas. El caso de uso terminará una vez realizada alguna de estas acciones.
Complejidad	Media
Prioridad	Media
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente. El usuario debe estar autenticado en el sistema.
Postcondiciones	La lista fue creada correctamente La lista fue editada correctamente La lista fue eliminada correctamente Las listas fueron visualizadas correctamente
Referencias	RF1, RF2, RF3 y RF4

Tabla 3: Descripción del caso de uso Eliminar ítem de una lista

Objetivo	Eliminar uno o varios documentos que contienen las listas.
Actores	Usuario (inicia): puede eliminar uno o varios documentos contenidos en las listas.
Resumen	El caso de uso inicia cuando el usuario selecciona en el menú la opción Mis listas, el sistema visualiza las listas (privadas y públicas) creadas por el usuario, este selecciona una lista, para realizar la acción de eliminar uno o varios documentos, el caso de uso finaliza con la realización de esta acción.

Complejidad	Media
Prioridad	Media
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente. El usuario debe estar autenticado en el sistema.
Postcondiciones	El ítem fue eliminado correctamente.
Referencias	RF5

Tabla 4: Descripción del caso de uso Crear referencias bibliográficas

Objetivo	Crear las referencias bibliográficas de los documentos seleccionados en un estilo bibliográfico determinado.
Actores	Usuario (inicia): crea las referencias bibliográficas de los documentos seleccionados escogiendo un estilo bibliográfico (IEEE, ISO-690, MLA, Harvard, Chicago, Turabian, Vancouver, APA).
Resumen	El usuario selecciona un estilo bibliográfico para crear las referencias bibliográficas de los documentos escogidos y la misma es mostrada en un panel en formato plano.
Complejidad	Media
Prioridad	Media
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente. El usuario debe estar autenticado en el sistema. La lista de documentos a referenciar debe estar creada.
Postcondiciones	Se crearon correctamente las referencias bibliográficas en el estilo bibliográfico escogido por el usuario.
Referencias	RF6

Tabla 5: Descripción del caso de uso Exportar referencias bibliográficas

Objetivo	Exportar los registros bibliográficos conformados por los metadatos de los documentos listados.
-----------------	---

Actores	Usuario (inicia): exporta las referencias bibliográficas a un formato de archivo determinado (HTML, XML, RTF, RIS, BibTeX, TXT).
Resumen	El usuario selecciona uno o varios documentos contenidos en una lista, y una vez creadas las referencias bibliográficas de dichos documentos, presiona la opción Exportar y puede seleccionar un formato de archivo determinado.
Complejidad	Media
Prioridad	Media
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente. El usuario debe estar autenticado en el sistema. La lista de documentos a exportar debe de estar creada. Las referencias bibliográficas que se deseen exportar deben estar creadas.
Postcondiciones	Se exportaron correctamente las referencias bibliográficas al formato de archivo seleccionado por el usuario.
Referencias	RF7

Tabla 6: Descripción del caso de uso Enviar referencias bibliográficas por correo electrónico

Objetivo	Enviar las referencias bibliográficas de los documentos seleccionados a la dirección de correo del usuario autenticado.
Actores	Usuario (inicia): envía las referencias bibliográficas de los documentos seleccionados a su dirección de correo.
Resumen	El usuario selecciona uno o varios documentos contenidos en una lista y luego presiona el botón Enviar.
Complejidad	Media
Prioridad	Baja
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente. El usuario debe estar autenticado en el sistema.

	La lista de documentos a enviar debe estar creada.
Postcondiciones	Se enviaron correctamente las referencias bibliográficas al correo electrónico del destinatario.
Referencias	RF8

2.5 Modelo de diseño

El diseño del sistema es otro flujo de trabajo que se realiza al final de la fase de elaboración. El modelo de diseño permite describir la realización física de los casos de uso, centrándose en el impacto que tienen en el sistema los requisitos funcionales y otras restricciones relacionadas con el entorno de implementación.

2.5.1. Diagrama de clases del diseño

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación (40). A continuación se muestran los diagramas por cada caso de uso.

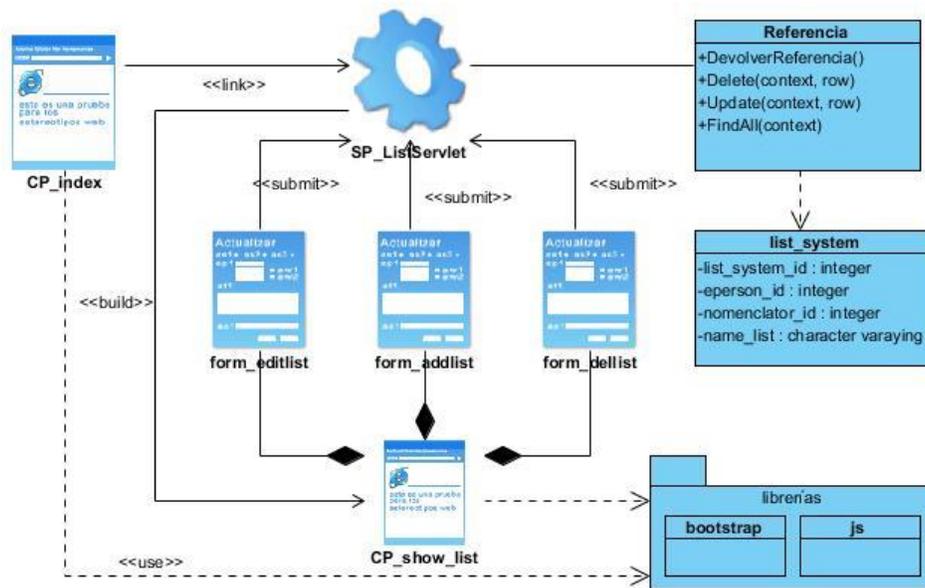


Figura 3: Diagrama de clases con estereotipos web del caso de uso Gestionar lista de documentos

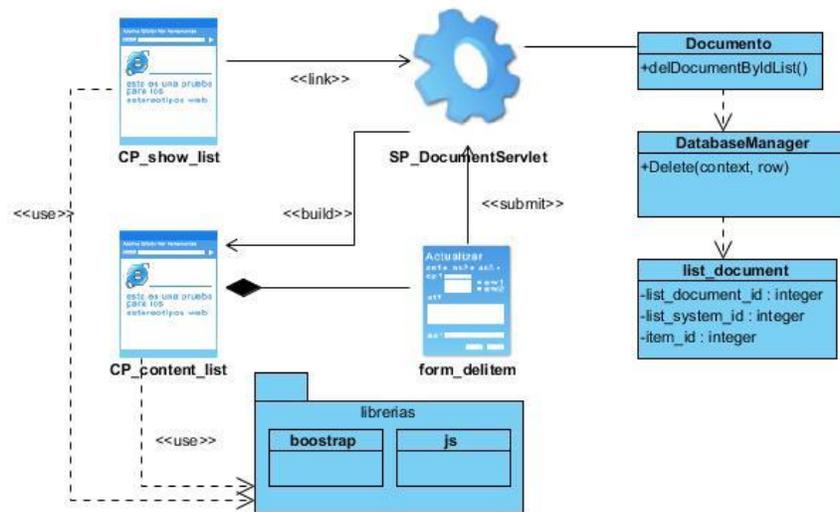


Figura 4: Diagrama de clases con estereotipos web del caso de uso Eliminar ítems de una lista

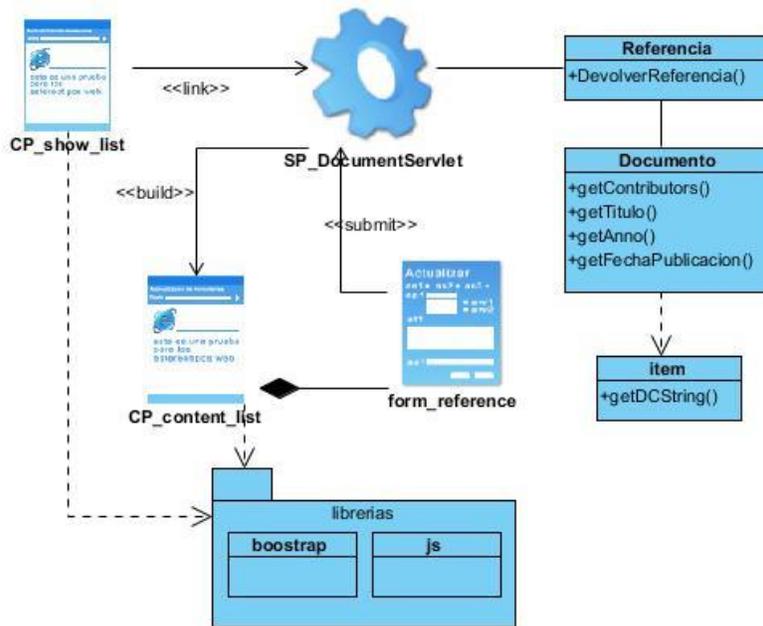


Figura 5: Diagrama de clases con estereotipos web del caso de uso Crear referencias bibliográficas

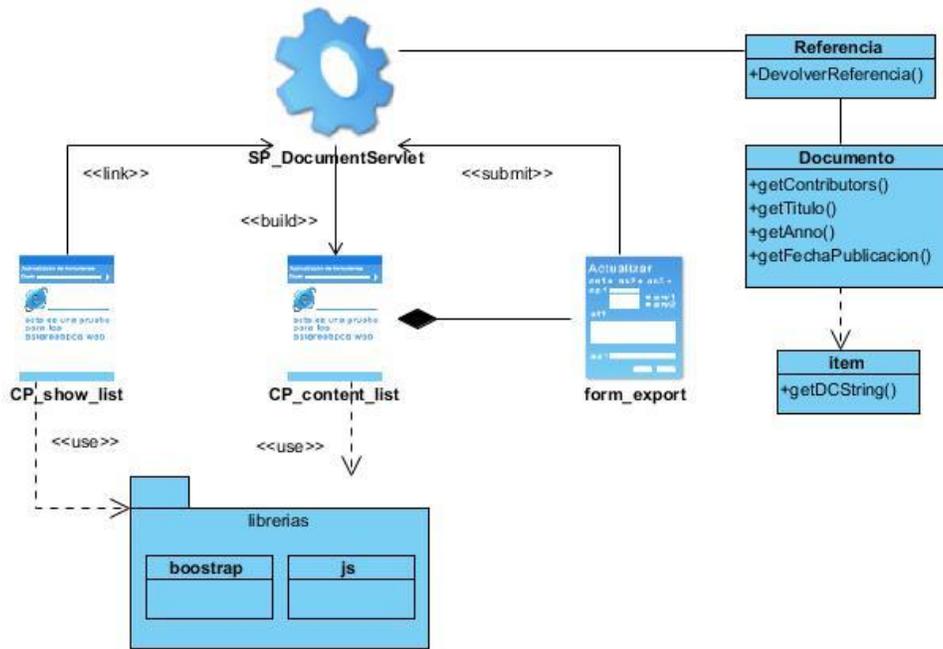


Figura 6: Diagrama de clases con estereotipos web del caso de uso Exportar referencias bibliográficas

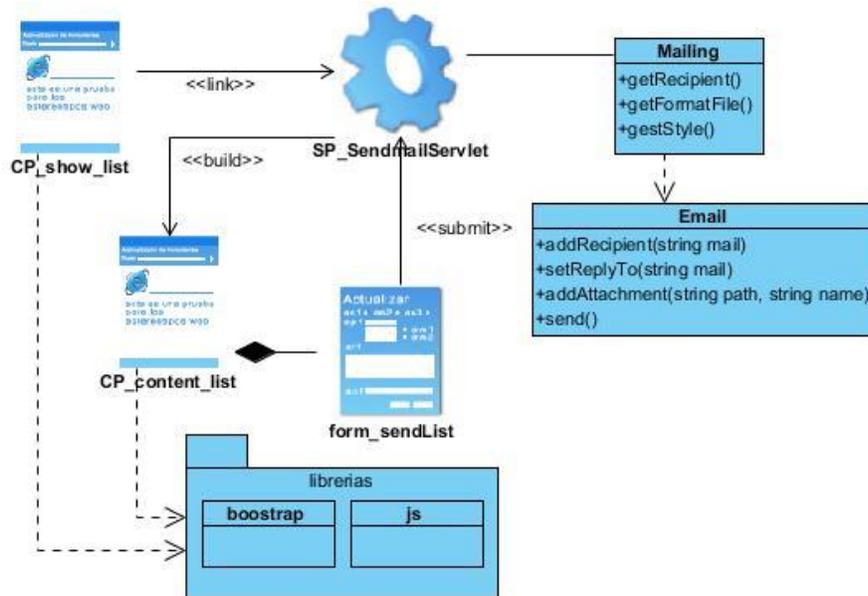


Figura 7: Diagrama de clases con estereotipos web del caso de uso Enviar referencias bibliográficas por correo electrónico

2.5.2. Diagrama de colaboración

Los diagramas de colaboración se utilizan para visualizar la interacción entre los objetos, en este caso es parte de la solución del módulo en desarrollo. Este tiene como objetivo la identificación de los requisitos y sus responsabilidades sobre los objetos. Las colaboraciones se organizan entorno a los

enlaces de unos objetos con otros (40). A continuación se muestran los diagramas de colaboración para los distintos casos de uso pertenecientes al módulo Referencias Bibliográficas.

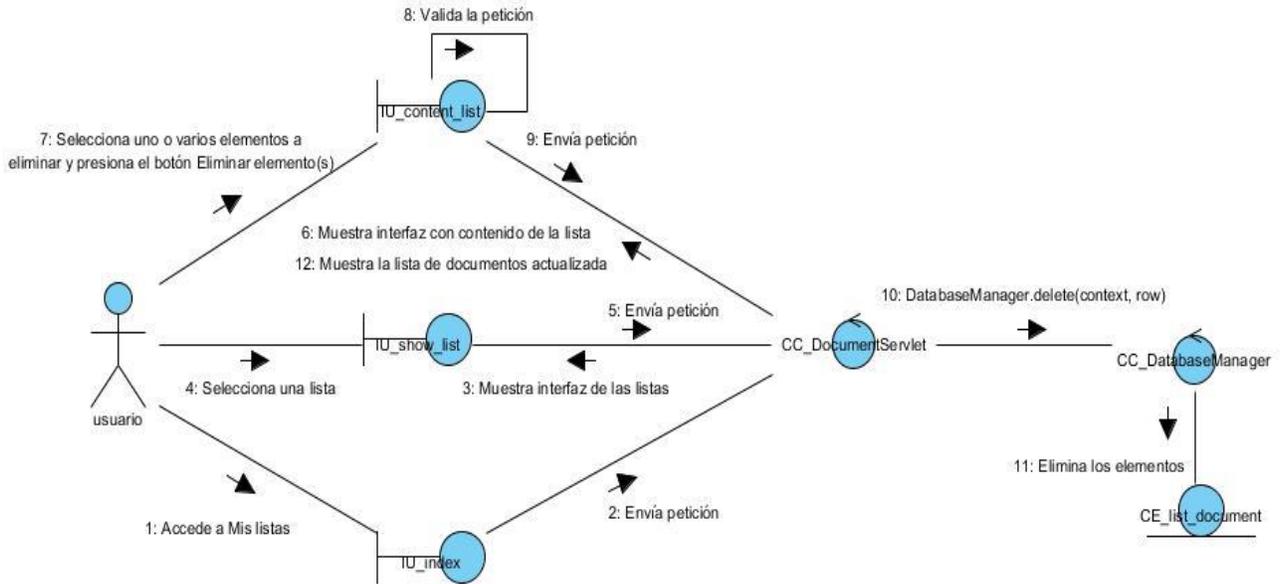


Figura 8: Diagrama de colaboración del requisito funcional Eliminar ítems de una lista

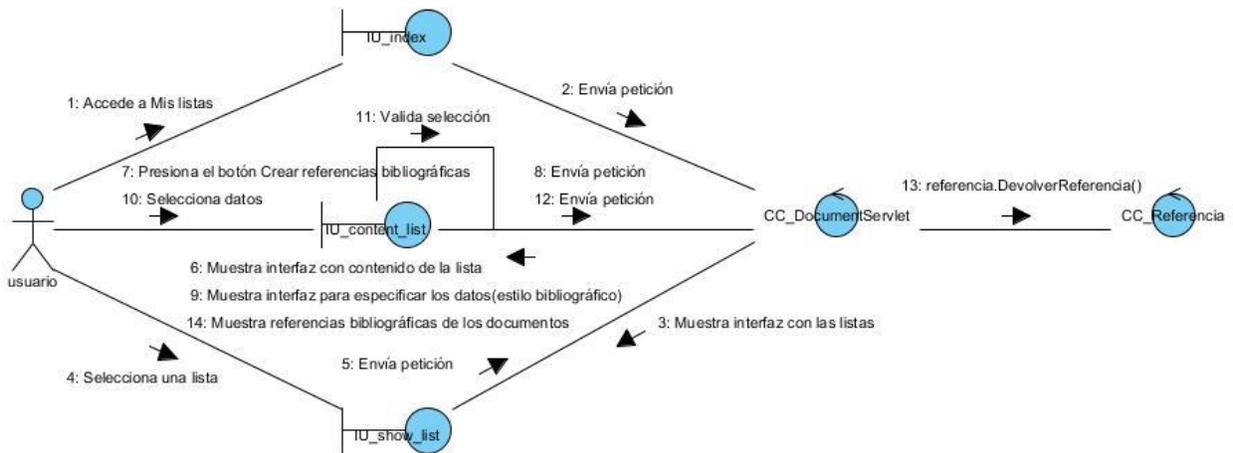


Figura 9: Diagrama de colaboración del requisito funcional Crear referencias bibliográficas

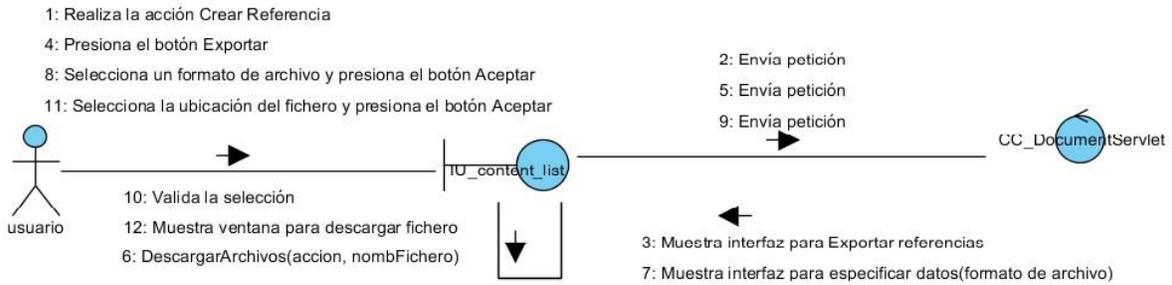


Figura 10: Diagrama de colaboración del requisito funcional Exportar referencias bibliográficas

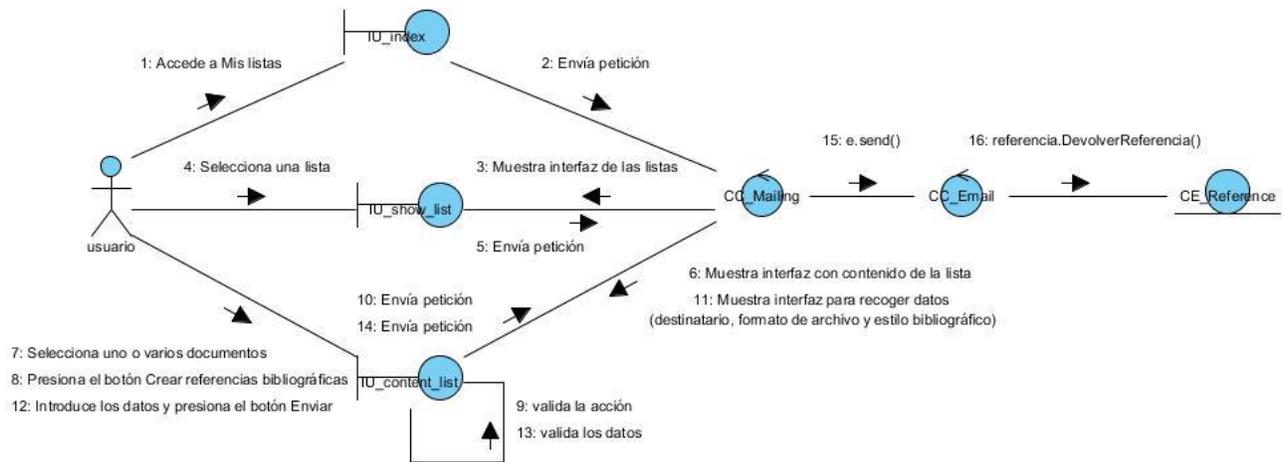


Figura 11: Diagrama de colaboración del requisito funcional Enviar referencias bibliográficas por correo electrónico

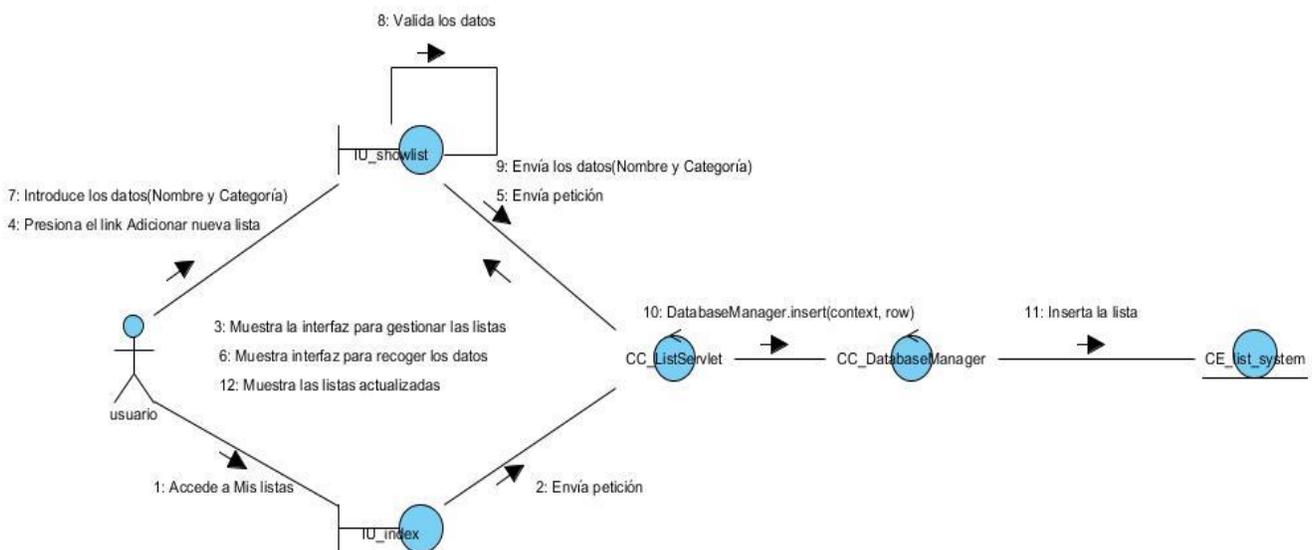


Figura 12: Diagrama de colaboración del requisito funcional Adicionar lista

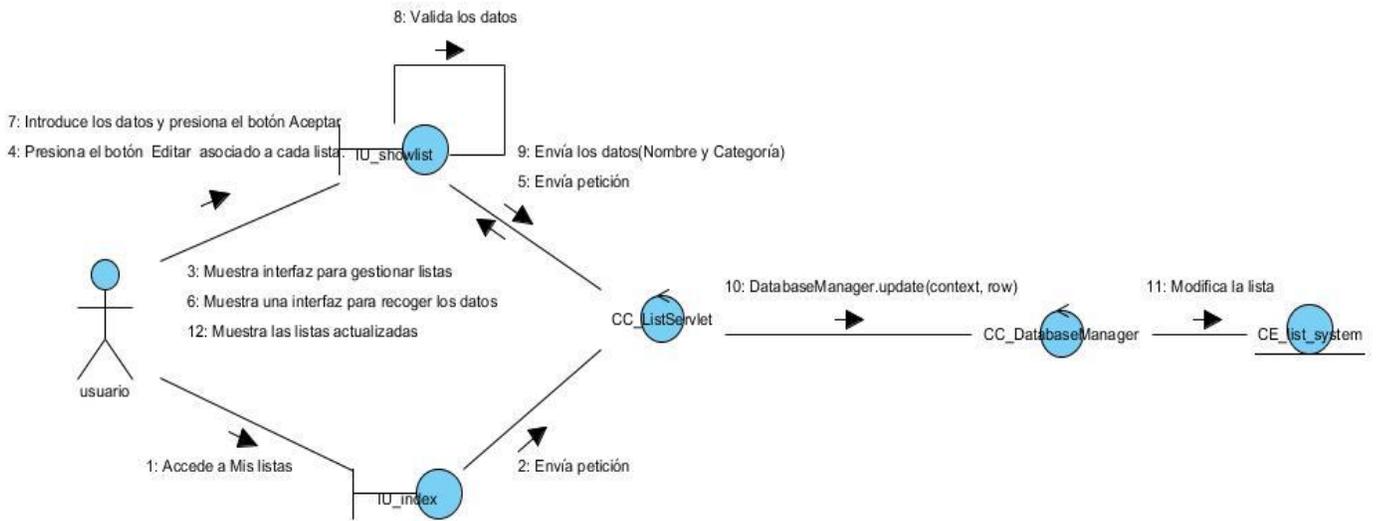


Figura 13: Diagrama de colaboración del requisito funcional Editar lista

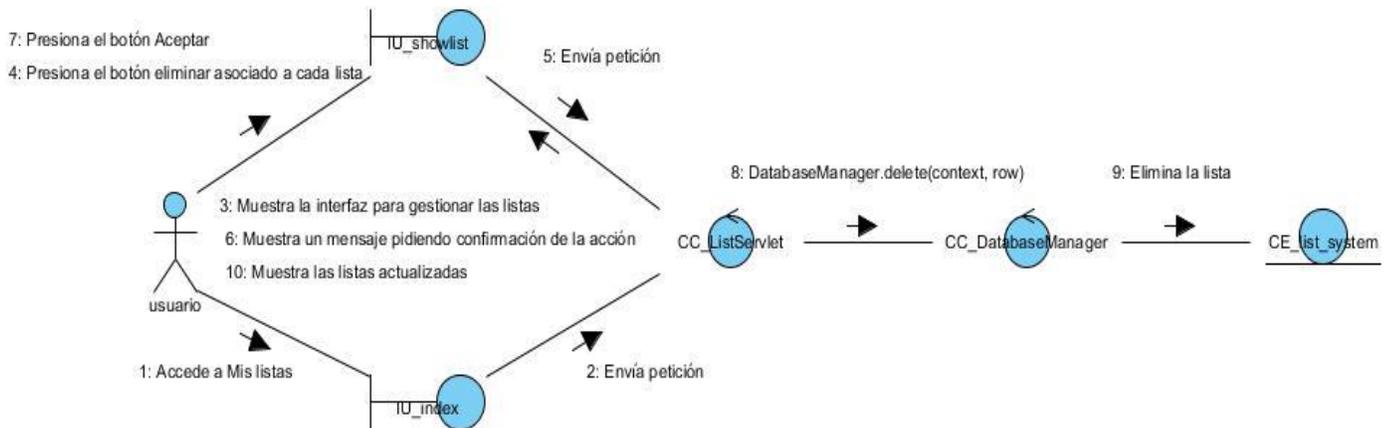


Figura 14: Diagrama de colaboración del requisito funcional Eliminar lista

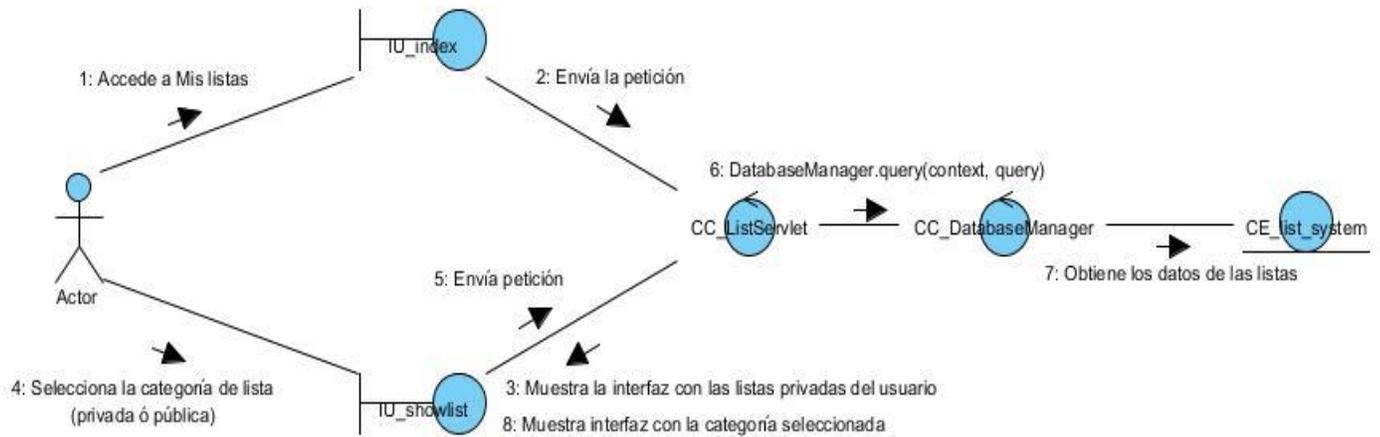


Figura 15: Diagrama de colaboración del requisito funcional Visualizar lista

2.5.3. Diagrama de paquetes

El diagrama de paquetes es una simplificación de un diagrama de clases, solo se representan los paquetes a los que pertenecen las clases. Además, permite organizar los elementos modelados con UML, facilitando de esta forma el manejo de los modelos de un sistema complejo.

Un paquete es un mecanismo utilizado para agrupar elementos de UML, pueden ser simples estructuras conceptuales o pueden estar reflejados en la implementación. Permiten dividir un modelo para agrupar y encapsular sus elementos en unidades lógicas individuales. Se pueden utilizar para plantear la arquitectura del sistema a nivel macro. (41)

A continuación se representa el diagrama de paquetes perteneciente al módulo Referencias bibliográficas del sistema REPXOS 3.0.

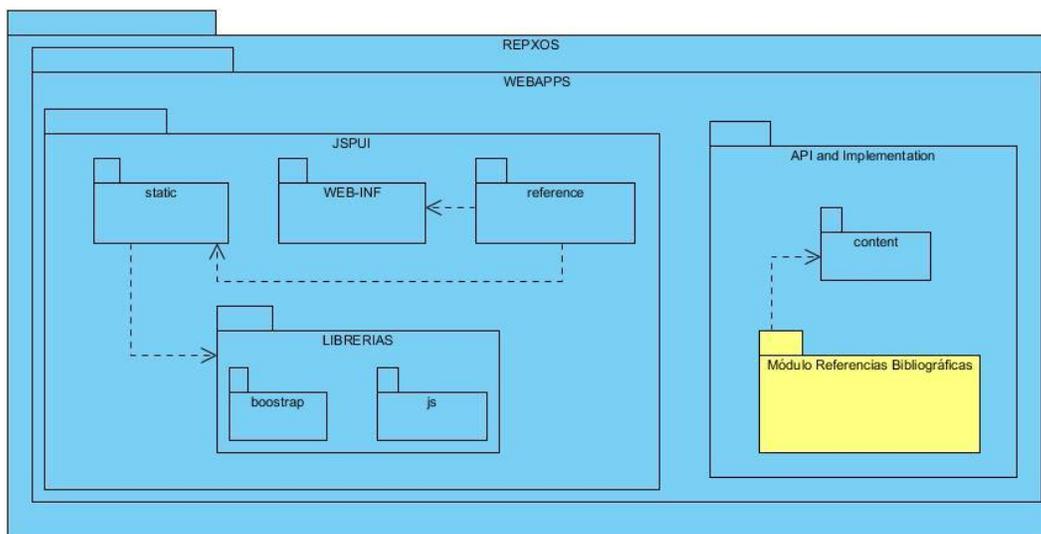


Figura 16: Diagrama de paquetes

Descripción del diagrama de paquetes

- **REPXOS:** paquete que agrupa al sistema REPXOS 3.0. En su interior se encuentran los módulos con que puede interactuar el usuario desde la interfaz web que se provee a este a través del navegador web.
- **WEBAPPS:** carpeta de despliegue donde se concentran los archivos esenciales para el funcionamiento de la aplicación web.
- **JSPUI:** reúne las interfaces web con las que interactúa el usuario a través del navegador web con la aplicación web, y paquetes de servlets que atienden las peticiones de los usuarios al sistema.
- **Static:** contiene los ficheros de hojas de estilo (css) y javascript utilizados por la aplicación web.
- **WEB-INF:** agrupa los ficheros de configuración local y de lenguaje sistema REPXOS 3.0.
- **Reference:** concentra las vistas del módulo Referencias Bibliográficas para el sistema REPXOS 3.0.
- **LIBRERIAS:** paquete que contiene los ficheros de las librerías de validación y estilos de las interfaces del módulo Referencias Bibliográficas.
- **Bootstrap:** agrupa los ficheros de hojas de estilo del framework de presentación Bootstrap utilizado por el sistema REPXOS 3.0 y las interfaces del módulo Referencias Bibliográficas.
- **Js:** contiene los ficheros de la librería JavaScript jQuery y componentes Bootstrap utilizados en las interfaces del sistema REPXOS 3.0 para la validación de los campos de entrada de datos.
- **API and Implementation:** agrupa las clases de la lógica del sistema REPXOS 3.0. Se subdivide en varios paquetes que manejan la seguridad, configuración y manejo de la información de las colecciones de documentos que pueden ser creadas con el sistema.
- **Content:** concentra los ficheros que manejan la lógica del sistema REPXOS 3.0 para subir documentos al repositorio y crear las colecciones de documentos en el mismo.
- **Módulo Referencias Bibliográficas:** concentra las clases que manejan la lógica del módulo referencias bibliográficas.

2.6 Modelo de datos físico

El modelo de datos es un conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructuras y tipos de datos, operaciones y restricciones. (4)

El modelo de datos físico es una representación de un diseño de los datos que consideren las instalaciones y los apremios de un sistema de gerencia dado, de base de datos. Un modelo físico

completo de los datos incluirá todos los artefactos de la base de datos requeridos para crear relaciones entre las tablas o para alcanzar metas del funcionamiento.

A continuación se muestra el modelo de datos físico del módulo Referencias Bibliográficas:

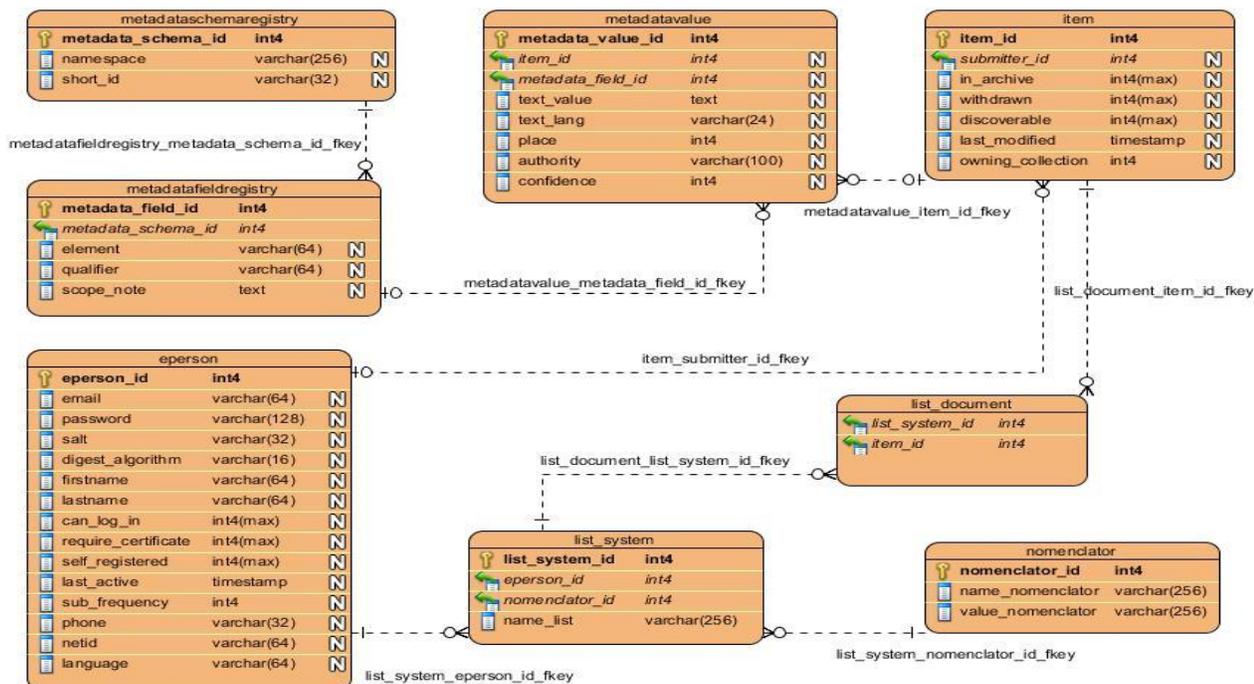


Figura 17: Modelo de datos del módulo Referencias bibliográficas

Metadataschemaregistry: contiene el nombre de dominio e id del esquema (dc, dcterms) de los metadatos.

Metadatafieldregistry: almacena el elemento y calificador a través de los cuales pueden ser obtenidos los metadatos de los documentos presentes en el repositorio.

Metadatavalue: contiene los metadatos de cada documento presente en el repositorio.

Eperson: almacena los datos de la persona registrada en el sistema.

Ítem: contiene datos como el id de la persona que subió el documento al repositorio, la fecha en que se realizó dicho envío y si se encuentra disponible el documento físico en el repositorio.

Nomenclator: almacena el tipo de nomenclador así como su nombre.

List_system: contiene el id de la persona que creó la lista, el id del nomenclador y el nombre que se con creó dicha lista.

List_document: contiene el id de la lista a la que pertenece el documento y el id del documento.

2.7 Arquitectura del sistema

La arquitectura de software se refiere a las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. El concepto de arquitectura de software se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad, modificabilidad), y servir como guía en el desarrollo. (42)

El sistema REPXOS 3.0 está basado en la herramienta para la creación de repositorios Dspace la cual utiliza el patrón arquitectónico de n capas, específicamente tres capas, lo que implica utilizar el mismo patrón de arquitectura para el módulo Referencias Bibliográficas.

La arquitectura en capas define el patrón en capas como una organización jerárquica. Lo que posibilita un diseño basado en niveles de abstracción creciente, posibilitando a los implementadores, particionar un problema en una secuencia de pasos incrementales. Este estilo de desarrollo en varios niveles, facilita que en caso que ocurra algún cambio, sólo se tendrían que realizar las correcciones necesarias en el nivel requerido sin tener que revisar código de otros niveles. La arquitectura está representada mediante un modelo en capas (Aplicación, Lógica de Negocio y Almacenamiento).

Capa de aplicación: en esta capa se diseña todo lo que constituye la interfaz gráfica y la interacción del usuario con el software. Se comunica únicamente con la capa de negocio. Representa el conjunto de componentes que genera la información que se representará en la interfaz de usuario del cliente.

La capa de aplicación se evidencia en el módulo mediante las interfaces que le son presentadas al usuario a través del navegador. Interfaces que representan a esta capa en el módulo Referencias Bibliográficas son show_list (interfaz principal del gestionar lista de documentos) y content_list (interfaz principal de contenido de una lista de documentos que reúne funcionales tales como: crear referencias bibliográficas y enviar vía e-mail).

Capa de lógica de negocio: en esta capa residen los objetos de negocio que interactúan con los objetos de dominio. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse para un correcto funcionamiento lógico de la aplicación. Los objetos de negocio, separan los datos y la lógica de negocio haciendo uso del modelo de objetos definido, dichos objetos de negocio son los encargados de procesar la información proveniente tanto de la Aplicación, como de la capa de Almacenamiento.

En el sistema, la **capa de lógica de negocio** se aprecia en el paquete **content** del módulo *API and Implementation* del código del sistema REPXOS 3.0, donde se agregaron las nuevas clases que suplen las funcionalidades del módulo Referencias Bibliográficas; ejemplo: Documento.java, Harvard.java, ListSystem.java, entre otras.

Capa de Almacenamiento: es el puente entre la capa de Lógica de Negocio y el Sistema de Base de Datos. Encapsula la lógica de acceso a datos. Aquí se encuentran componentes que hacen transparente el acceso a la base de datos. Este es el lugar idóneo para implementar los objetos de acceso a datos, permitiendo ingresar, obtener, actualizar y eliminar información del Sistema de Bases de Datos. Los Objetos de Acceso a Datos (DAOs) encapsulan la persistencia de los objetos de dominios, proveen la persistencia de los objetos transitorios y las actualizaciones de los objetos existentes en la base de datos. Las implementaciones de los DAOs estarán disponibles para los objetos de negocio.

La capa de Almacenamiento está representada en el módulo Referencias Bibliográficas por los objetos creados en las clases Documento.java y ListSystem.java de la clase DatabaseManager.java para poder ejecutar las consultadas necesarias a través de los métodos implementados en esta última.



Figura 18: Arquitectura de tres capas en el módulo Referencias Bibliográficas

2.1 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Su principal objetivo es incrementar la calidad del software en términos de reusabilidad, mantenimiento y extensibilidad. (43)

Los patrones del diseño tratan los problemas del diseño que se repiten y que se presentan en situaciones particulares del diseño, con el fin de proponer soluciones a ellas. Por lo tanto, los patrones de diseño

son soluciones exitosas a problemas comunes. Existen muchas formas de implementar patrones de diseño. (44)

Los patrones de diseño utilizados para la implementación del módulo Referencias Bibliográficas son:

Patrón Experto: es el encargado de asignar una responsabilidad al experto en información, es decir la clase que cuenta con la información necesaria para cumplir con la misma. Esto permite que los sistemas tiendan a ser más fáciles de entender, mantener y ampliar; y además presenta la oportunidad de reutilizar los componentes en futuras aplicaciones. (45)

Este patrón se pone de manifiesto en el módulo al momento de la construcción de las clases controladoras (Documento.java, ListSystem.java, etc.), encargadas de obtener los datos en las vistas de gestión de las listas y el contenido de las mismas

Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. Su propósito fundamental es encontrar un creador que debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. (46)

Al momento de crear una referencia bibliográfica este patrón se pone de manifiesto en el módulo, ya que le permite al servidor mediante la clase Referencia.java construir en dependencia del estilo seleccionado por el usuario en la vista de la funcionalidad, el objeto encargado de devolver la cadena con los datos del (los) documento (s) en cuestión.

Data Access Object (DAO): suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos tales como una Base de Datos o un archivo (47). En otras palabras, proporciona una capa independiente en donde centraliza todo el acceso a datos, de forma tal que los aísla del resto de la aplicación.

En el módulo Referencias Bibliográficas se crean DAOs como DatabaseManager que es responsable de ejecutar las consultas a la base de datos necesarias para la obtención de los datos a mostrar en el sistema. Además se evidencia en el momento de acceder a la vista de gestión de listas (públicas y privadas) y a la de contenidos de una lista, cuando en el servidor se ejecutan consultas para obtener las listas registradas en la base de datos y los documentos que pertenecen a las mismas.

Bajo Acoplamiento: se encarga de asignar responsabilidades de manera que el acoplamiento permanezca bajo. El acoplamiento es una medida de la fuerza con que un elemento está conectado a, tiene conocimiento de y confía en, otros elementos. Un elemento con bajo (o débil) acoplamiento no depende demasiado de otros elementos, además soporta clases más independientes. Todo esto

propicia que no se afecten cambios en otros componentes, sea fácil de entender y conveniente para reutilizar. (48)

Al momento de enviar un correo electrónico con el archivo de las referencias bibliográficas, la clase controladora de esta funcionalidad, solo crea los objetos necesarios para poder realizar la operación. Además las clases contenidas en el módulo Referencias Bibliográficas fueron implementadas de forma independientes, esto mitiga el impacto de los cambios que se puedan realizar en una clase.

Alta Cohesión: asigna una responsabilidad de modo que la cohesión siga siendo alta. La alta cohesión permite que la clase que tiene una responsabilidad pueda interactuar con otras clases para poder realizar las tareas encomendadas. (45)

Este patrón se evidencia en el módulo en el momento de acceder a la funcionalidad Mis listas, atendida por el servletlist el cual provee solo la instancia necesaria para el manejo de dicha información.

Singleton: garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella, restringiendo la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. (49)

Este patrón está presente en cada funcionalidad del módulo, debido a que las clases de lógica del negocio atienden al criterio de crear únicas instancias para las operaciones que se realicen en el momento.

Conclusiones del capítulo

En el presente capítulo se desarrolló la propuesta de solución del sistema, obteniéndose a partir de un análisis las funcionalidades que debe tener el módulo, lo que ofreció una mejor comprensión del mismo. La confección del modelo de dominio permitió mostrar todas las clases conceptuales utilizadas en la solución. Con la realización del diagrama de casos de uso se logró mostrar la relación entre los actores y los casos de uso del sistema. Para guiar la creación de la herramienta se definió el patrón arquitectónico tres Capas y los patrones de diseño. La confección del diagrama de clases del diseño, permitió especificar correctamente las clases de software que se deben utilizar para dar paso a la implementación, y visualizar las relaciones entre las clases que involucran el sistema. Además, con la realización del modelo de entidad-relación se lograron representar las tablas de la base de datos que se utilizaron.

CAPÍTULO 3. Implementación y pruebas

En este capítulo se describen los elementos necesarios para la implementación, partiendo del resultado obtenido del diseño. Se muestra la distribución del sistema en nodos mediante el diagrama de despliegue y la organización de los componentes y las relaciones lógicas entre ellos a través del diagrama de componentes, quedando así conformado el modelo de implementación. Se exponen las pruebas realizadas a cada caso de uso, siguiendo particularmente, el método de pruebas de Caja Negra.

3.1 Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes, cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros.

3.1.1. Estándares de codificación

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad, es de gran importancia para la calidad del software y para obtener un buen rendimiento. Además, si se aplica de forma continuada un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas, y, posteriormente, se efectúan revisiones del código de rutinas, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y de mantener. (50)

La forma de escribir código es propia de cada programador y completamente diferente a la forma de cualquier otro. De la forma usada depende la facilidad para entender el código y retomar ciertas partes realizadas por otros integrantes, así como la depuración de las mismas.

A continuación se enumeran las normas a seguir en la codificación, junto con ejemplos, así como las justificaciones de los mismos (51):

1. Clases servlets:

- 1.1. Comenzar el nombre de la clase en letra inicial mayúscula; en caso de estar compuesto por más de una palabra, la inicial de esta también debe estarlo. Al final del nombre de la clase siempre debe aparecer la palabra Servlet. No se utilizan delimitadores () para discernir entre palabras. Ejemplo: DocumentServlet.java.
- 1.2. Las variables que identifiquen IDs deben comenzar con letra inicial minúscula y terminar con la palabra Id. No se utilizan delimitadores () para discernir entre palabras. Ejemplo: ítemId.

- 1.3. Los nombres de variables temporales deben comenzar con letra inicial minúscula; en caso de estar compuesto por más de una palabra, la inicial de esta debe aparecer en mayúscula. No se utilizan delimitadores () para discernir entre palabras. Ejemplo: docsList.
 - 1.4. Las variables que capturen datos enviados desde la capa de aplicación deben nombrarse igual al valor del atributo **name** del componente web que envió dicha información. Ejemplo: nbList.
 - 1.5. Los nombres de los métodos de estas clases deben comenzar con letra inicial minúscula; en caso de estar compuesto por más de una palabra, la inicial de esta debe aparecer en mayúscula. No se utilizan delimitadores () para discernir entre palabras. Ejemplo: doGet().
2. Clases de la capa de negocio:
- 2.1. Comenzar el nombre de la clase en letra inicial mayúscula; en caso de estar compuesto por más de una palabra, la inicial de esta también debe estarlo. No se utilizan delimitadores () para discernir entre palabras. Ejemplo: Documento.java.
 - 2.2. Los nombres de variables referentes a los metadatos de los documentos deben comenzar con letra inicial minúscula; en caso de estar compuesto por más de una palabra, la inicial de esta debe aparecer en mayúscula. Ejemplo: ciudadPublicacion.
 - 2.3. Los nombres de variables temporales deben comenzar con letra inicial minúscula; en caso de estar compuesto por más de una palabra, la inicial de esta debe aparecer en mayúscula. No se utilizan delimitadores () para discernir entre palabras.
 - 2.4. Los nombres de los métodos de estas clases deben comenzar con letra inicial minúscula; en caso de estar compuesto por más de una palabra, la inicial de esta debe aparecer en mayúscula. Deben ser sugerentes a la función que realizan y estar en idioma inglés. No se utilizan delimitadores () para discernir entre palabras. Ejemplo: delDocumentById().

3.1.2. Diagrama de componentes

Los diagramas de componentes representan todos los tipos de elementos de software que intervienen en el desarrollo de aplicaciones informáticas y sus dependencias (compilación y ejecución). Estos diagramas se utilizan para modelar la vista estática de un sistema y muestran la organización y dependencias lógicas entre un conjunto de componentes de software, sean estos componentes de código fuente, librerías, binarios o ejecutables. Un componente es una parte física y reemplazable de un sistema que se conforma con un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usan para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interfaces. (40)

El paquete de la solución para el trabajo con las referencias bibliográficas está dividido en tres subsistemas de implementación fundamentales: el subsistema de la Aplicación, el subsistema de Lógica

de Negocio y el subsistema de Almacenamiento, estructurados de forma tal que se agrupan de acuerdo con el papel que desempeñan dentro del patrón arquitectónico de tres Capas.

A continuación se representa el diagrama de componentes del módulo Referencias Bibliográficas de forma global:

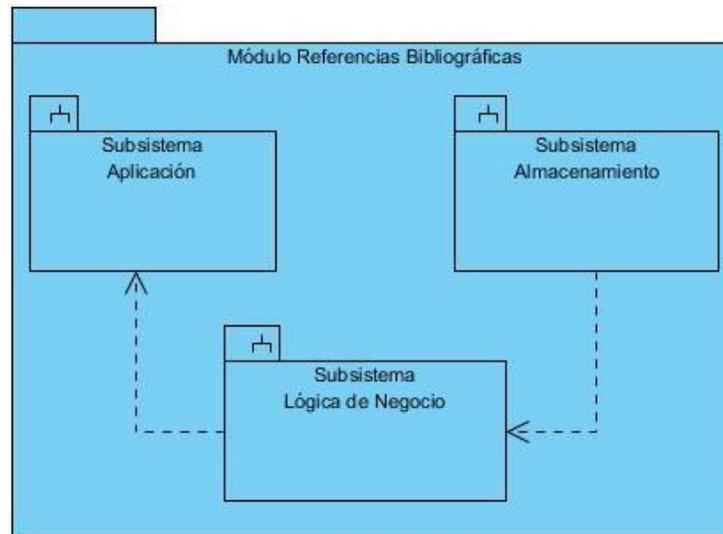


Figura 19: Diagrama de componentes del módulo Referencias Bibliográficas

El subsistema Aplicación contiene los componentes necesarios para la interacción del usuario con el sistema, los cuales son manejados por el subsistema de Lógica de Negocio. A continuación se representan de forma detallada los componentes que integran dicho subsistema:

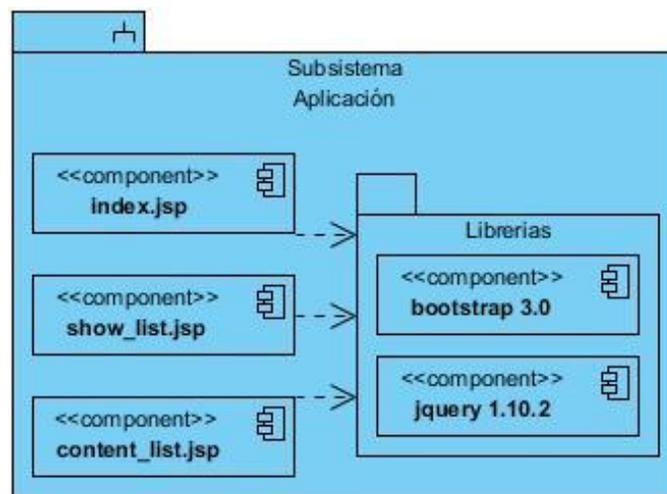


Figura 20: Diagrama de componentes del subsistema Aplicación

El subsistema del negocio, es el rector de las actividades de la aplicación, este contiene los ficheros de código fuente, los cuales interactúan con los demás subsistemas coordinando las acciones del software. A continuación se representa de forma detallada los componentes que integran dicho subsistema:

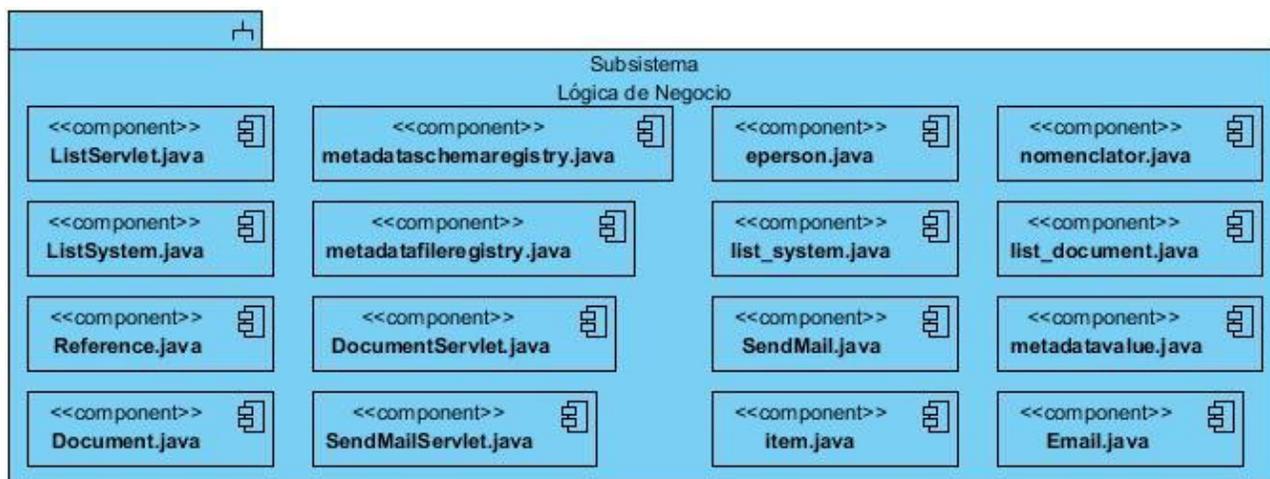


Figura 21: Diagrama de componentes del subsistema negocio

El subsistema del Almacenamiento es el encargado de la interacción con la base de datos, para de esta forma gestionar la información con la que trabaja el sistema. A continuación se representa de forma detallada los componentes que integran dicho subsistema:

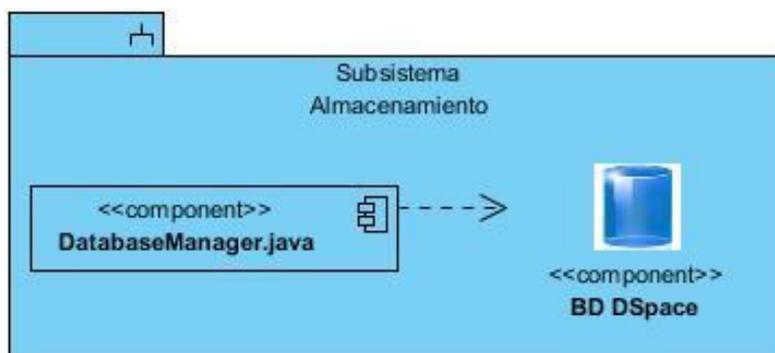


Figura 22: Diagrama de componente del subsistema Acceso a Datos

3.1.3. Diagrama de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño (19).

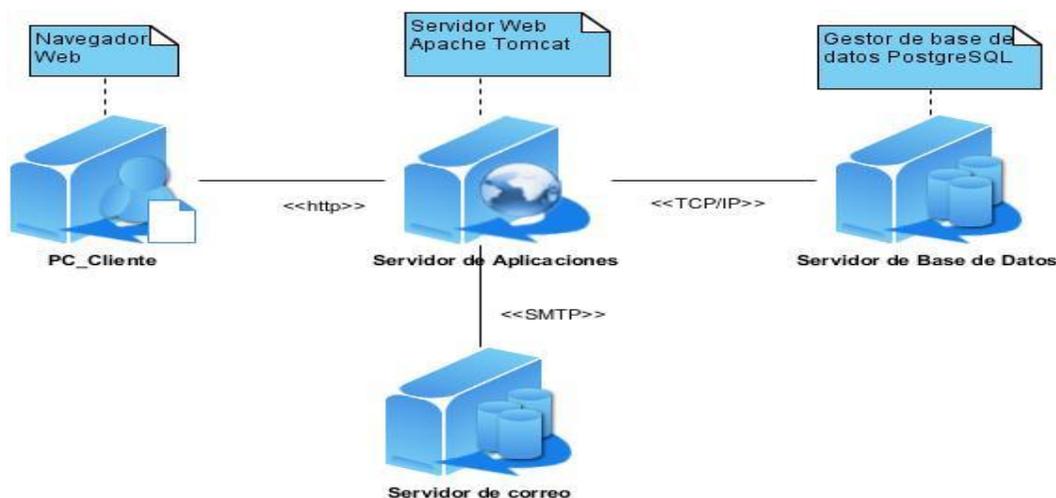


Figura 23: Diagrama de despliegue del módulo Referencias Bibliográficas

El diagrama de despliegue refleja la disposición de los recursos necesarios para realizar el despliegue del sistema. A continuación se describen características de los nodos:

- **Nodo PC_Cliente:** representa las computadoras que utilizarán los usuarios para interactuar con la aplicación. Se comunica con el Servidor de Aplicaciones a través del protocolo HTTP.
- **Nodo Servidor de Aplicaciones Web Apache Tomcat:** representa el servidor donde se encuentra instalado el sistema.
- **Nodo Servidor de Base de Datos PostgreSQL:** es el servidor PostgreSQL donde se almacena la base de datos del sistema.
- **Nodo Servidor de Correo:** representa el servidor de correo que se requiere para el envío de las referencias bibliográficas a un determinado usuario. La entidad determina qué servidor de correo desea utilizar. Realiza el envío del correo electrónico mediante el protocolo SMTP por el puerto 25.

3.2 Pruebas

Realizar pruebas a una aplicación desarrollada es una actividad fundamental que toda empresa productora de software debería llevar a cabo. Las pruebas de software son un elemento clave en la garantía de la calidad del producto final.

Uno de los mayores problemas que se afrontan actualmente en la esfera de la informática es la calidad del software, por lo que el proceso de pruebas es sin dudas uno de los aspectos fundamentales para medir el estado de calidad de un sistema informático.

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados. Los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente (40).

3.3 Estrategia de prueba

La estrategia de prueba proporciona la descripción de los pasos que hay que llevar a cabo como parte de la prueba, cuándo se deben planificar y realizar esos pasos, y cuánto esfuerzo, tiempo y recursos se van a requerir (52). Incluye los niveles de prueba (unidad, integración, etc.) a ser diseccionados, el tipo de prueba a ser ejecutada (funcional, estrés, etc.) y los casos de prueba diseñados para lograr los objetivos. Además integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados, que dan como resultado una correcta construcción del software.

3.3.1. Niveles de prueba

La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. La construcción del software se realiza siguiendo un plan de desarrollo que comienza por la implementación de componentes individuales con interfaces bien definidas. Posteriormente, estos componentes de software se van integrando con otros de forma incremental hasta obtener el producto completo. A las pruebas se realizadas sobre componentes, módulos o métodos individuales se le denomina **pruebas unitarias** o **pruebas modulares**. Al integrar los diferentes componentes pueden surgir errores en la interacción que son el objetivo de las **pruebas de integración**. Sobre el software ya completo se realizan **pruebas de sistemas** y, tras estas, se instala en su entorno de operación. Para evaluar la adecuación del sistema a dicho entorno, se ejecutan **pruebas de implantación**. Finalmente el producto está preparado para ser evaluado por el cliente y se realizan **pruebas de aceptación** con el objetivo de saber si se puede poner el software en producción. (53)

Durante todo el desarrollo, la identificación de defectos y su corrección conllevan la necesidad de reproducir las pruebas ya realizadas, con el objetivo de asegurar que no se han introducido nuevos errores en el funcionamiento. A estas pruebas se le denomina **pruebas de regresión**. (53)

A continuación se describen los niveles de prueba que serán utilizados en el módulo Referencias Bibliográficas del sistema REPXOS 3.0:

- **Pruebas de aceptación:** estas pruebas se realizan para que el cliente certifique que el sistema es válido para él. La planificación detallada de estas pruebas debe haberse realizado en etapas tempranas del desarrollo del proyecto, con el objetivo de utilizar los resultados como indicador

de su validez: si se ejecutan las pruebas documentadas a satisfacción del cliente, el producto se considera correcto y, por tanto, adecuado para su puesta en producción.

- **Pruebas de sistema:** son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados.

3.3.2. Tipos de prueba

- **Funcionalidad**

Entre los tipos de pruebas que se realizan en un sistema, está el tipo que evalúa la funcionalidad de éste. Cada tipo de prueba tiene un objetivo específico y una técnica que lo soporta: (54)

- Prueba de Función
- Prueba de Seguridad
- Prueba de Volumen

Al módulo Referencias Bibliográficas para el sistema REPXOS 3.0 se le aplica la prueba de funcionalidad específicamente la de **Función**.

Prueba de Función (54)

Objetivo: asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

Metas: verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio así como la apropiada aceptación de datos.

Enfoque: los requisitos funcionales (Casos de Uso) y las reglas del negocio.

Técnica: Caja Negra.

- **Rendimiento**

Las pruebas de rendimiento son las pruebas que se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. Están diseñadas para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado.

Este tipo de prueba incluye las siguientes pruebas:

- Pruebas de contención
- Pruebas de profile
- Pruebas de carga

Al módulo Referencias Bibliográficas para el sistema REPXOS 3.0 se le aplica la prueba de Rendimiento específicamente la de **Carga**.

Pruebas de Carga

Este es el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación.

Para medir el rendimiento de la aplicación se utilizó la herramienta **Apache JMeter 2.13**, que a continuación se explica:

Apache JMeter es una herramienta de carga diseñada para realizar Pruebas de Rendimiento y Pruebas Funcionales sobre Aplicaciones Web. Está diseñado para desarrollar diferentes tipos de test; permitiendo diseñar tanto sencillos tests que soliciten simples páginas web, como complejas secuencias de requisiciones que permitan evaluar el comportamiento de una aplicación o como la capacidad de carga máxima que pueda tener una aplicación en un servidor (pudiendo llegar a saturar el servidor). Utilizar JMeter en aplicaciones web para la comprobación de los recursos del sistema, supone una mayor efectividad en el proceso y en la fiabilidad de los resultados. (55)

Esta herramienta genera un Informe Agregado donde muestra alguno de los resultados necesarios para el análisis de la prueba de carga.

El **Informe Agregado** crea una fila por cada petición en el test. Por cada una de estas filas se muestra la siguiente información:

- Etiqueta: El nombre de la muestra (conjunto de muestras).
- # Muestras: El número de muestras para cada URL.
- Media: El tiempo medio transcurrido para un conjunto de resultados.
- Mín: El mínimo tiempo transcurrido para las muestras de la URL dada.
- Máx: El máximo tiempo transcurrido para las muestras de la URL dada.
- % Error: Porcentaje de las peticiones con errores.
- Rendimiento: Rendimiento medido en base a peticiones por segundo/minuto/hora.
- Kb/sec: Rendimiento medido en Kilobytes por segundo.

A continuación se exponen los resultados de las pruebas aplicadas al módulo Referencias Bibliográficas del sistema REPXOS 3.0:

- **Simulación de 200 usuarios concurrentemente**

Para la ejecución de la prueba, se introducen 200 hilos (número de usuarios a simular) y 1 segundo como período de subida (tiempo que debiera llevarle a JMeter lanzar todos los hilos), lo que va a simular

la concurrencia para 200 usuarios. Como resultado se obtiene que el módulo Referencias Bibliográficas es capaz de soportar esa cantidad de usuarios concurrentemente, esto deducido de la columna representativa del tanto por ciento de errores para cada una de las peticiones asociadas a cada conjunto de muestras (que es 0.0% para todos los casos), para las prestaciones con las que se hizo la prueba (en este caso las mismas que se utilizaron para el desarrollo del módulo). Con menos prestaciones no se garantiza el mismo comportamiento. Los resultados arrojaron además que el rendimiento se comportó con una media de 27.8 peticiones por segundo y la latencia (entendida como el tiempo de espera para la renderización de la página, el tiempo en obtener respuesta del servidor) para cada conjunto de pruebas no supera el valor de 6453 milisegundos (6.453 segundos).

A continuación se muestra el Informe Agregado asociado a la prueba de carga realizada al módulo Referencias Bibliográficas:

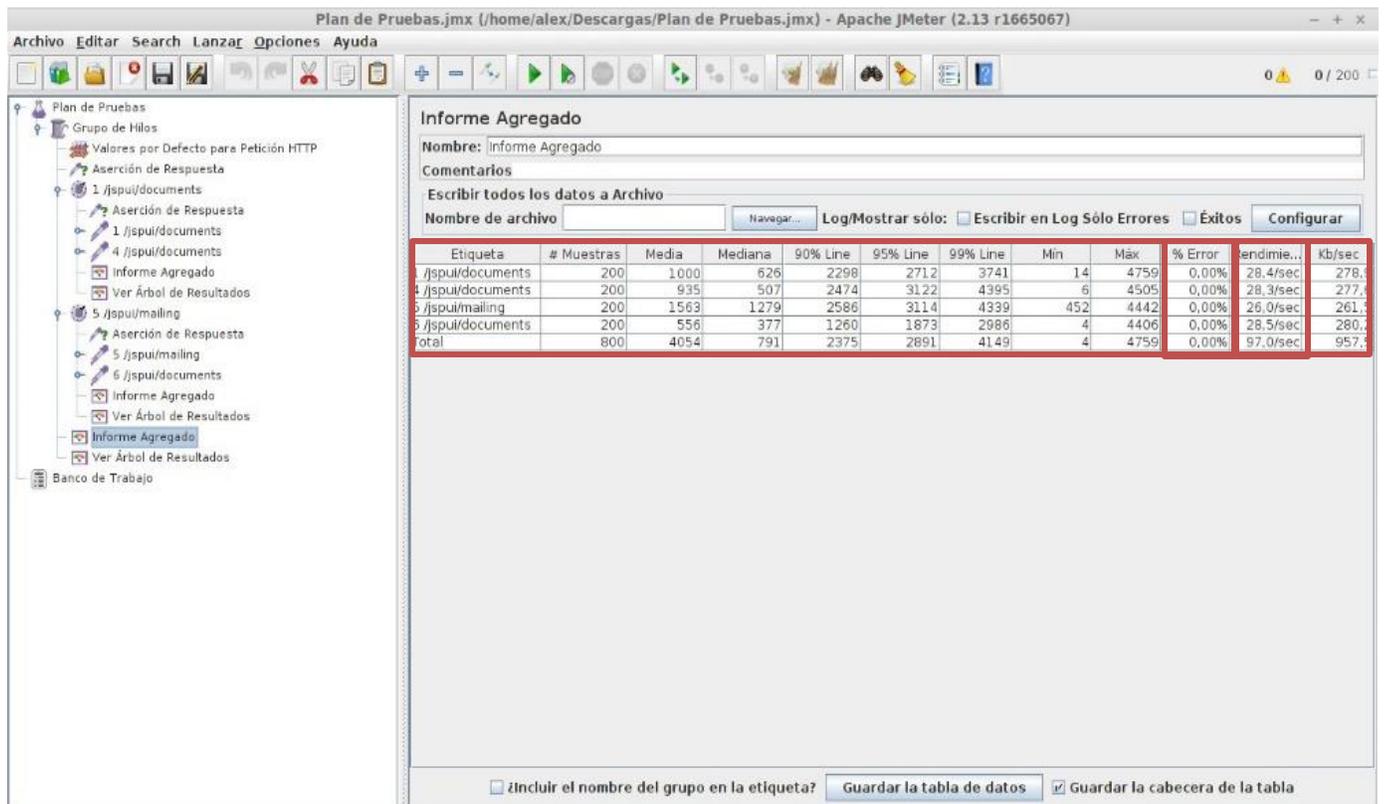


Figura 24: Informe Agregado de la prueba de carga al módulo Referencias Bibliográficas

3.3.3. Método de prueba

- **Pruebas de caja negra:** obtienen casos a partir de los requisitos funcionales del sistema a probar, para demostrar que las funciones son operativas, por lo que no se tiene en cuenta la forma en que se codifica esa funcionalidad, sino que se consideran únicamente las entradas y las salidas. (52)

Las pruebas de caja negra pretenden encontrar estos tipos de errores:

- Funciones incorrectas o ausentes
- Errores en la interfaz
- Errores en estructuras de datos o en accesos a bases de datos externas
- Errores de rendimiento
- Errores de inicialización y de terminación

3.3.4. Técnica de prueba

Para desarrollar las pruebas de caja negra existen varias técnicas, algunas de ellas son (29):

- Técnica de la Partición de Equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del Análisis de Valores Límites: prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

La técnica que se va a utilizar en el módulo Referencias bibliográficas del sistema REPXOS 3.0 es la de **partición de equivalencia**.

- **Partición de equivalencia**

Esta técnica de prueba de caja negra divide el dominio de entrada de un programa en clases de datos, a partir de las cuales deriva los casos de prueba. Cada una de estas clases de equivalencia representa a un conjunto de estados válidos o inválidos para las condiciones de entrada. (56)

3.3.5. Diseño de casos de prueba

El diseño de casos de prueba es una parte de las pruebas de componentes y sistemas en las que se diseñan los casos de prueba (entradas y salidas esperadas) para probar el sistema. El objetivo de este proceso de diseño es crear un conjunto de casos de pruebas que sean efectivos descubriendo defectos en los programas y muestre que el sistema satisface sus requerimientos. (57)

A continuación se muestra un ejemplo de un caso de prueba diseñado para el módulo Referencias Bibliográficas del sistema REPXOS 3.0:

DCP Crear Referencias bibliográficas

Descripción general: Permite crear las referencias bibliográficas de los documentos seleccionados en un estilo bibliográfico determinado y la misma es mostrada en un panel en formato plano.

Condiciones de ejecución: El usuario debe estar autenticado en el sistema.

Tabla 7: Prueba de caja negra del caso de uso Crear referencias bibliográficas

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Crear referencias bibliográficas correctamente.	Permite crear las referencias bibliográficas de los documentos seleccionados en un estilo bibliográfico determinado correctamente.	Muestra un panel con las referencias bibliográficas conformadas en formato plano.	<ol style="list-style-type: none"> 1. Seleccionar la opción Mis listas del menú principal. 2. Seleccionar una lista de documentos. 3. Seleccionar uno o varios elementos de la lista de los que desee obtener las referencias bibliográficas. 4. Presionar el botón Crear referencias bibliográficas. 5. Seleccionar un estilo bibliográfico.

Descripción de las variables:

Tabla 8: Descripción de las variables del caso de uso Crear referencias bibliográficas

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Título del documento	Campo de selección múltiple	No	Permite seleccionar uno o varios documentos para realizar la acción de crear las referencias bibliográficas
2	Estilo bibliográfico	lista desplegable	No	En este campo se seleccionan los estilos bibliográficos. Brinda las opciones APA, IEEE, ISO-690, Turabian, Harvard, MLA, Chicago y Vancouver.

3.4 Resultado de las pruebas

A continuación se muestran los resultados de las pruebas de caja negra que fueron realizadas al módulo Referencias Bibliográficas, en las que se ejecutaron un total de dos iteraciones. Se observa además el total de no conformidades encontradas y las que fueron resueltas, e igualmente se muestra la cantidad que no procedieron.

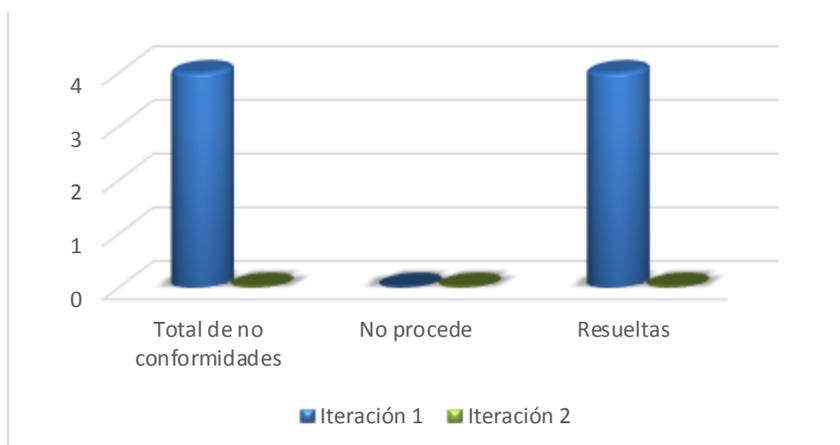


Figura 25: Resultados de las pruebas de caja negra aplicadas al módulo Referencias Bibliográficas del sistema REPXOS 3.0

En la primera iteración se detectaron un total de cuatro no conformidades, en su mayoría validación de datos en las diferentes interfaces. De ellas fueron resueltas las cuatro correctamente. En la segunda iteración no se encontraron no conformidades, validándose de esta manera el correcto funcionamiento del módulo Referencias Bibliográficas para el sistema REPXOS 3.0.

Conclusiones del capítulo

En el presente capítulo se realizó el diagrama de componentes, que permitió mostrar las dependencias entre éstos, y visualizar con más facilidad la estructura general del sistema, así como el comportamiento del servicio que estos componentes proporcionan. Con la realización del diagrama de despliegue se logró especificar la plataforma sobre la que se ejecuta el software del sistema. La definición de los estándares de codificación permitió mejorar el entendimiento del código perteneciente al Módulo Referencias Bibliográficas del sistema REPXOS 3.0 por otros desarrolladores, y alcanzar una uniformidad en el mismo. Trazando la estrategia de prueba se definieron una serie de pasos bien planificados que dan como resultado una correcta construcción del sistema, entre los que se incluyen los tipos de prueba que se le realizaron al software. La aplicación del método de caja negra, permitió validar correctamente los requisitos funcionales del sistema utilizándose para ello la técnica de partición de equivalencia.

Conclusiones

Con la realización de este trabajo:

- Se realizó una fundamentación teórica con los conceptos significativos del dominio de la investigación, así como de sistemas similares, logrando un mejor entendimiento de los procesos de extracción y conversión de metadatos de documentos para la generación de referencias bibliográficas.
- Se caracterizaron las tecnologías y herramientas definidas por el proyecto permitiendo determinar donde serían utilizadas efectivamente para el desarrollo de la solución propuesta.
- Se propuso una solución, la cual permitió tener una concepción más clara de lo que se debía desarrollar y se especificaron las funcionalidades que tenía que cumplir el módulo Referencias Bibliográficas.
- Se diseñó la solución, permitiendo a los desarrolladores tener la base necesaria para realizar de forma exitosa la implementación del sistema.
- Se logró desarrollar una solución para los usuarios que accedan al Sistema para repositorios digitales REPXOS 3.0, y que necesiten obtener las referencias bibliográficas de los documentos consultados, siendo estas referencias utilizadas para sus trabajos de investigación, o importadas a un gestor bibliográfico.
- Se diseñaron las pruebas para comprobar el correcto funcionamiento del módulo, y se utilizaron para validar el mismo, con lo que se obtuvieron resultados satisfactorios, comprobándose que el módulo Referencias Bibliográficas cumple con todas las funcionalidades especificadas correctamente.

Por todo lo antes planteado se puede concluir que con el desarrollo de este trabajo de diploma se dio cumplimiento al objetivo general propuesto.

Recomendaciones

Una vez culminado el trabajo se propone como recomendación, continuar el desarrollo de este módulo, incorporándole una funcionalidad que permita generar las referencias bibliográficas de un documento que no esté contenido en alguna lista, teniendo en cuenta las nuevas tecnologías que se desarrollan en la actualidad y de donde se implante el sistema.

Referencias Bibliográficas

1. **J. A. Dávila, L. A. Núñez.** Repositorio Institucional de la Universidad de los Andes. [En línea] octubre de 2005. [Citado el: 19 de febrero de 2015.] http://www.saber.ula.ve/bitstream/123456789/33588/1/repositorios_intitucionales.pdf.
2. **Mayans, Concepción Díaz.** Referencias bibliográficas estilo Vancouver. *Centro de Nuevas Tecnologías de la Información y las Comunicaciones*. [En línea] [Citado el: 17 de enero de 2015.] <http://www.cpimtz.sld.cu/normvanc.htm>.
3. **Castilla, Liuris Rodríguez.** ¿Le resulta difícil hacer la bibliografía? Los gestores de referencias bibliográficas pueden ayudarlo. *Scientific Electronic Library Online. SciELO*. [En línea] febrero de 2009. [Citado el: 19 de febrero de 2015.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352009000200003.
4. **Senso, José A. y de la Rosa Piñero, Antonio.** Scielo(Scientific Electronic Library Online). *El concepto de metadatos. Algo más que recurso electrónico*. [En línea] 2013. [Citado el: 3 de diciembre de 2014.] <http://ria.asturias.es/RIA/bitstream/123456789/344/1/citas%20y%20referencias.pdf>.
5. **Universidad Nacional de Colombia.** SIAMAC Sistema de Información de la Amazonia Colombiana. *Metadatos*. [En línea] [Citado el: 2 de abril de 2015.] www.unal.edu.co/siamac/sig/metadatos1.html.
6. **Universidad de Granada.** UGR Universidad de Granada. Campus de Excelencia Internacional. *Metadatos*. [En línea] 2006. [Citado el: 2 de abril de 2015.] www.ugr.es/~afporcel/metadatos.doc.
7. **Testa, Patricia y Ceriotto, Paula.** *Descripción de objetos digitales: metadatos*. Buenos Aires, Argentina : s.n., 2009.
8. **Saburit, Arelis Borrell.** Centro de ayuda Biblioteca Virtual de Salud de Cuba. *Gestores Bibliográficos*. [En línea] 2012. bvsayuda.sld.cu/ayudas/tutorial/bfle-resulta-dificil-hacer-la-bibliografia-los-gestores-de-referencias-bibliograficas-pueden-ayudarlo/gestores-bibliograficos.
9. **Universidad Distrital Francisco José de Caldas.** *Gestores bibliográficos*. [En línea] [Citado el: 4 de mayo de 2015.] <http://sistemadebibliotecas.udistrital.edu.co/index.php/servicios/recursos-bibliograficos/gestores-bibliogr%C3%A1ficos>.
10. **Universidad de Salamanca.** Biblioteca Universidad de Salamanca de España. *Gestores Bibliográficos*. [En línea] 2015. <http://bibliotecas.usal.es/mendeley>.
11. **Universidad de Zaragoza.** Biblioteca Universidad de Zaragoza España. *Guía de Mendeley*. [En línea] [Citado el: 13 de marzo de 2015.] http://biblioteca.unizar.es/archivos_biblio/19/Guia_Mendeley.pdf.
12. **Ruiz, Diego José.** Universidad de Murcia España. *Gestores de referencia: EndNote web*. [En línea] [Citado el: 18 de febrero de 2015.]
13. **Universidad politécnica de Valencia.** *Descripción y Acceso a RefWorks*. [En línea] 2012. [Citado el: 22 de febrero de 2015.]
14. **Universidad de Alcalá.** Biblioteca Universidad de Alcalá. *Gestores Bibliográficos*. [En línea] 2013. [Citado el: 22 de febrero de 2015.]

15. **Biblioteca de la Facultad Iberoamericana de Ciencias Sociales.** Mexico. *Estilos bibliográficos*. [En línea] 2014. [Citado el: 30 de enero de 2015.]
16. **Murrillo, Jacqueline.** NISABA. *Estilos bibliográficos*. [En línea] 2011. [Citado el: 3 de abril de 2015.]
17. **Biblioteca universitaria de Córdoba.** *Gestores de referencia. Citas y referencias*. [En línea] [Citado el: 21 de enero de 2015.] <http://www.uco.es/servicios/biblioteca/gestorref/estilos-enw.pdf>.
18. **Universidad Oberta de Catalunya.** Biblioteca de la Universidad Oberta de Catalunya. *Ayuda del repositorio Institucional*. [En línea] [Citado el: 3 de mayo de 2015.] http://openaccess.uoc.edu/webapps/o2/help/index_es.html#about.
19. **Pérez, Tomás Saorán.** Tesis Doctoral de la Universidad de Murcia. *Modelo conceptual para la automatización de bibliotecas en el contexto digital*. [En línea] 2002. [Citado el: 3 de mayo de 2015.] <https://digitum.um.es/xmlui/bitstream/10201/84/3/SaorinPerez.pdf.txt>.
20. **Tecnológico de Monterrey.** Biblioteca Campus Cuernavaca. *Misión*. [En línea] <http://www.cva.itesm.mx/biblioteca/>.
21. **Universidad de Málaga.** Biblioteca de la Universidad de Málaga. *Google Académico*. [En línea] [Citado el: 2 de mayo de 2015.] http://riuma.uma.es/xmlui/bitstream/handle/10630/7299/Guia_Google%20Academico%202014.pdf?sequence=1.
22. **Kruchten, Philippe.** *The Rational Unified Process: An Introduction*. Boston - Estados Unidos : Addison-Wesley, 2000. ISBN: 0201707101.
23. **Alvarez, Jorge C.** Universidad de Cartagena. *Metodologías de desarrollo de Software RUP*. [En línea] 2012. [Citado el: 19 de diciembre de 2014.]
24. **Fabri, F.** Softonic, La guía de software más completa del mundo. . *Kit esencial para compilar y ejecutar tus programas Java*. [En línea] 2015. <http://java-development-kit-jdk.softonic.com/>.
25. **Open Courses of Atlantic International University.** . *El lenguaje de programación Java*. . [En línea] [Citado el: 18 de enero de 2015.] <https://cursos.aiu.edu/Lenguajes%20de%20Programacion/PDF/Tema%202.pdf>.
26. **Flanagan, David.** Dialnet . *JavaScript. La Guía Definitiva*. [En línea] 2007. [Citado el: 1 de febrero de 2015.] ISBN: 978-84-415-2202-2 84-415-2202-2.
27. **Pardo, Otto Colomina.** Tema 4: Frameworks JavaScript. JQuery. Universidad de Alicante. *Tecnologías Web*. [En línea] 2011. <http://hdl.handle.net/10045/19909>.
28. **Vallecillo, Antonio y Fuentes, Lidia.** Revista Novatica–Asociación de Técnicos de Informática-España. *Una introducción a los perfiles UML*. [En línea] 2004. [Citado el: 20 de enero de 2015.]
29. **Quesada, Juan Antonio López.** Universidad de Murcia. Departamento de informática y sistemas. *Lenguajes de marcas y sistemas de gestión de información*. [En línea] 2014. [Citado el: 22 de febrero de 2015.] http://dis.um.es/~lopezquesada/documentos/IES_1415/LMSGI/curso/material.html#ut5.
30. **Pressman, Roger S.** *Ingeniería de Software, un enfoque práctico. Quinta edición*. s.l. : McGraw-Hill Companies, 2002. ISBN: 8448132149.

31. **Gairín, Jose Manuel Rodríguez.** Facultad de Biblioteconomía y Documentación Universidad de Barcelona. *DSpace: un manual específico para gestores de la información y la documentación*. [En línea] 2008. [Citado el: 26 de febrero de 2015.]
32. **Maven.** Apache Maven Project. *Maven 3.0*. [En línea] <https://maven.apache.org/docs/3.0/release-notes.html>.
33. **Sitio Web Oficial Apache Tomcat.** *Apache Tomcat*. [En línea] [Citado el: 15 de febrero de 2015.] <http://tomcat.apache.org/>.
34. **Hatcher, Erik y Loughran, Steve.** *Java Development with Ant. (1st edition)*. [En línea] Agosto de 2002. ISBN 978-1930110588..
35. **Gutiérrez, Javier J.** Universidad de Sevilla . *¿Qué es un framework web?* [En línea] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
36. **Mestras, Juan Padrón.** Universidad Complutense Madrid. *Bootstrap 3.0 Aplicaciones Web/Sistemas Web*. [En línea] [Citado el: 3 de marzo de 2015.]
37. **Rivera, Javier González.** Aula de software libre de la universidad de Córdoba. *Maqueta DCC PostgreSQL*. [En línea] 6 de junio de 2014. [Citado el: 22 de febrero de 2015.]
38. **Lozano, Mishel Carolina Rueda.** Escuela de ingeniería en sistemas y computación. *Diseño de una página web de una tienda virtual mediante la herramienta de netbeans*. [En línea] enero de 2013. [Citado el: 20 de marzo de 2015.]
39. **Espinoza, Gleidis Y. Rosabal.** *Especificación de requisitos de software Proyecto repositorio Institucional*. La Habana : Universidad de las Ciencias Informáticas, 2015.
40. **Jacobson, Ivar, Booch, Grady and Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000.
41. **Gutierrez, Damian.** Universidad de los Andes. *UML, Diagrama de paquetes*. [En línea] Septiembre de 2009. [Citado el: 15 de abril de 2015.] http://www.codecompiling.net/files/slides/UML_clase_05_UML_paquetes.pdf.
42. **L. Bass, P. Clements, R. Kazman.** *Software Architecture in Practice, 2nd Edition*. s.l. : Addison Wesley, 2003.
43. **Ureta, Luis Alfredo Becerra.** University of the Basque Country UPV/EHU. [En línea] Patrones de diseño. [Citado el: 13 de abril de 2015.]
44. **Villa, Madelys Cuesta.** Repositorio Institucional de la Universidad de las Ciencias Informáticas. *Modelo para la ayuda a la toma de en la selección de patrones de desarrollo*. [En línea] Mayo de 2007. [Citado el: 15 de mayo de 2015.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_0261_07/1/TD_0261_07.pdf.
45. **Lerman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999. ISBN/970-17-0261-1.
46. **Visconti, Marcello y Astudillo, Hernán.** Universidad Técnica federico Santamaría. *Fundamento de Ingeniería de Software*. [En línea] [Citado el: 16 de mayo de 2015.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.

47. **Mesa, José Luis.** Biblioteca de la Universidad del Valle. *Patrón DAO*. [En línea] [Citado el: 15 de mayo de 2015.] eisc.univalle.edu.co/materias/Material_Desarrollo_Software/expoDAO.pdf.
48. **Martínez, Ivette C.** Laboratorio Docente de Computación. Universidad Simón Bolívar. *Diseño con patrones*. [En línea] [Citado el: 16 de mayo de 2015.] <http://ldc.usb.ve/~ci3715/teoria/patrones.pdf>.
49. **Obe, Regina y Hsu, Leo.** *PostGIS in action*. Stanford : Manning Publications Co., 2011. ISBN: 978-1-935182269.
50. **Microsoft Developer Network.** *Revisiones de código y estándares de codificación*. [En línea] 2015. [Citado el: 5 de mayo de 2015.] <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
51. **Fernández, Luis Carlos Álvarez.** *Estándares de codificación para Java Repositorio Institucional (RI)*. La Habana : Universidad de las Ciencias Informáticas, 2013.
52. **Universidad nacional abierta y a distancia.** Ingeniería de Software. *Capítulo 9. Estrategias de Prueba del Software*. [En línea] [Citado el: 22 de abril de 2015.]
53. **Tuya, Javier, Román, Isabel Ramos y Cosín, José Javier Dolado (ed.).** *Técnicas cuantitativas para la gestión en la ingeniería del software*. s.l. : NetBiblo, 2007. ISBN>978-84-9745-204-5..
54. **Blank, Isabel.** Universidad Simón Bolívar. Ingeniería de Software 2. Dpto. de Computación y Tecnología de la Información). *Pruebas de funcionalidad*. [En línea] Mayo de 2005. [Citado el: 26 de abril de 2015.]
55. **Sociedad Informática del Gobierno Vasco.** *Apache JMeter. Manual de usuario*. [En línea] [Citado el: 30 de mayo de 2015.]
56. **Universidad de Almería.** Universidad de Almería. Dpto. de lenguajes y Computación. *Técnicas de prueba*. [En línea] [Citado el: 13 de abril de 2015.]
57. **Sommerville, Ian.** *Ingeniería del software*. . s.l. : Pearson Educación, 2005.
58. **Franco., Andrés Cordón.** Departamento de Ciencias de la Computación e Inteligencia Artificial UNIVERSIDAD DE SEVILLA. *Sistema de Gestión de base de datos*. [En línea] 2006. [Citado el: 22 de febrero de 2015.]

Glosario de términos

ANSI: Sistema de codificación de caracteres alfanuméricos diseñado en el American National Standards Institute, que permite hasta 256 caracteres distintos.

Apache Software Foundation: es una organización no lucrativa creada para dar soporte a los proyectos de software bajo la denominación Apache, incluyendo el popular servidor HTTP Apache.

BibTeX: BibTeX usa un formato de archivo basado en texto e independiente del estilo para definir listas de elementos bibliográficos, como artículos, libros, tesis. Los archivos de bibliografía de BibTeX usualmente terminan en `.bib`. Los ítems bibliográficos incluidos en un `.bib` están separados por tipos.

Dublin Core: es un modelo de metadatos elaborado y auspiciado por la DCMI (Dublin Core Metadata Initiative), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos y a promover el desarrollo de los vocabularios especializados de metadatos para describir recursos para permitir sistemas más inteligentes el descubrimiento del recurso.

Módulo: En programación un módulo es una porción de un programa de computadora. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso).

Index Medicus: era un índice bibliográfico completo de artículos de revistas científicas centradas en los campos de ciencias médicas, publicados desde 1879 hasta 2004. Fue iniciado por John Shaw Billings, director de la Biblioteca de la Oficina del Cirujano General, Ejército de Estados Unidos. Esta biblioteca más tarde se convirtió en la Biblioteca Nacional de los Estados Unidos de Medicina (NLM).

ISI Web of Knowledge: es un servicio en línea de información científica, suministrado por el Instituto para la Información Científica (ISI por sus siglas en inglés), grupo integrado en Thomson Reuters. Facilita el acceso a un conjunto de bases de datos bibliográficas y otros recursos que abarcan todos los campos del conocimiento académico.

Java Server Pages (JSP): es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java

Marco de trabajo: Es un agregado de componentes orientado a dar solución a un tipo de problemas, sirve como referencia para desarrollar aplicaciones informáticas, brinda un conjunto amplio de funcionalidades lo que implica ahorro de tiempo en la implementación de sistemas informáticos.

PubMed: es un sistema de búsqueda desarrollado por el Centro Nacional de Información sobre Biotecnología (NCBI por sus siglas en inglés) en la Biblioteca Nacional de Medicina (NLM por sus siglas en inglés) de EE.UU. para integrar la información bibliográfica y factual de interés en Ciencias de la Salud. Permite el acceso a bases de datos bibliográficas compiladas por la NLM

RIS: Un archivo RIS (.ris) es un archivo de texto plano (similar a un .txt) que puede contener datos de una o de varias referencias bibliográficas. Los datos tienen una codificación y una estructura, es decir un formato. RIS es un formato que se acepta en todos tipo de software bibliográfico para exportar e importar datos.

XML: siglas en inglés de *eXtensible Markup Language* ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. XML se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.