

Universidad de las Ciencias Informáticas

Facultad 2



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

Componente para Adquisición a través del Protocolo OAI-PMH en el Sistema ABCD 3.0

Autor(a): Maryeris Esperance Aliaga

Tutor(es): Ing. Luis Carlos Alvarez Fernández
Ing. Reynier Carbonell Sánchez

La Habana, 21 de junio de 2018
“Año 59 de la Revolución”

Pensamiento



“No se trata de sueños ni fantasías; comienzan a ser visibles realidades. ¡Bien vale la pena luchar por ellas!”

Fidel Castro Ruz

Declaración de autoría

Declaro ser autora de la presente tesis que tiene por título: “Componente para Adquisición a través del protocolo OAI-PMH en el Sistema ABCD 3.0” y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Maryeris Esperance Aliaga

Firma del Autor

Ing. Luis Carlos Alvarez Fernández

Firma del Tutor

Ing. Reynier Carbonell Sánchez

Firma del Tutor

Datos de contacto

Tutores:

Ing. Luis Carlos Alvarez Fernández (lcalvarez@uci.cu).

Graduado de Ingeniero en Ciencias Informáticas en el año 2010. Se desempeña actualmente como Jefe de Departamento de Desarrollo de Componentes en el Centro de Informatización de la Gestión Documental (CIGED). Tiene 8 años de experiencia.

Ing. Reynier Carbonell Sánchez (reyniercs@uci.cu).

Graduado de Ingeniero en Ciencias Informáticas en el año 2014. Desempeña actualmente el rol de desarrollador en el proyecto para la Automatización de Bibliotecas y Centros de Documentación (ABCD). Tiene 4 años de experiencia en el desarrollo de software.

Agradecimientos

Agradezco con todo mi corazón a mi mamá Nancy, quien más que madre ha sido una amiga y confidente, quien me ha dado fuerzas para seguir adelante y quien sin desistir jamás me ha guiado siempre con paciencia, perseverancia y comprensión, en todos mis pasos, para hacer realidad mis sueños.

A mi papá Marcos, por haber confiado siempre en mí, por su apoyo, amor, cariño y dedicación, por hacerme entender que no importa las veces que caiga en la vida, sino las veces que sepa levantarme, por haber estado a mi lado en los momentos más difíciles.

A mi abuela María, por ser la mejor abuela del mundo, por quererme, consentirme y malcriarme tanto. A mi madrastra Consuelo, quien junto a mi papá me educó, me aconsejó e hizo de mí una mejor persona. A mi “hermana” Yisela, con quien he compartido muchos momentos de tristeza y felicidad.

Gracias a mi hermano Jorge, mi prima Ailén y mi tía Anarda, quien es como mi segunda madre, siempre linda y risueña y para mí ha sido toda su alegría y optimismo. A toda mi familia, por haber contribuido directamente con mi logro.

Gracias al amor de mi vida, mi Pichurri, por haberme brindado siempre su apoyo incondicional, por su ternura y pasión y por haber estado para mí cuando más lo necesité. Te amo hoy y siempre.

A mi suegra Mary, que aún con su carácter complicado y fuerte siempre me cantaba una canción que solo tenía 3 letras: “Tienes que estudiar”.

Gracias de corazón a mis tutores Reynier y Luis Carlos, por haberme guiado en el desarrollo de la tesis, por haber sido los más implicados en la realización de la

misma y por el apoyo que me brindaron siempre. A ellos debo la satisfacción del deber cumplido.

Le doy gracias a todos los amigos que hice en esta etapa de mi vida, a los nuevos, a los viejos Joel, el Moro, a la primera persona que conocí en la UCI, “Elizabeth”, te quiero mucho. A quienes compartieron conmigo alegría, felicidad y tristeza. Dany, la Yune, Antonio, Felipe, Alianhs, Alejandro, Maikel, a todos los pepillos y pepillas del aula, los quiero.

En fin, gracias a todas las personas que de una forma u otra hicieron posible este logro, que es tanto mío como de todos ustedes.

Dedicatoria

Dedico mi tesis a las personas más importantes de mi vida, mis padres, que han sido amigos, aliados y confidentes; sin ustedes, así de tiernos y enfurecidos como saben ser, no habría sido posible este logro. Si alguna vez no les he dicho cuánto los amo, este es el momento en el que deben saber, que todo lo que he sido y en lo que me he convertido es fruto del empeño y dedicación que siempre han sentido por mí.

Los amo con todo mi corazón.

A mi Pichurri, por haber aguantado mi mal carácter durante los 4 años que llevamos juntos y aún así, solo supo darme amor, cariño, comprensión y ánimo para seguir adelante.

Te amo.

Resumen

La interoperabilidad en los actuales Sistemas Integrados de Gestión de Bibliotecas (SIGB) se ha convertido en una necesidad, pues posibilita la comunicación entre ellos y que los mismos puedan compartir datos, documentos, entre otros, de forma efectiva. El advenimiento e incremento de las Tecnologías de la Información y las Comunicaciones (TIC) han propiciado que las bibliotecas y los centros de documentación satisfagan las necesidades de los usuarios de adquirir conocimientos e información, así como la informatización de estos centros. En la Universidad de las Ciencias Informáticas (UCI) existe el Centro de Informatización de la Gestión Documental (CIGED), el cual cuenta con varios proyectos, entre ellos, el Sistema para la Automatización de Bibliotecas y Centros de Documentación versión 3.0 (ABCD v3.0) y el Sistema para Repositorios y Bibliotecas Digitales versión 3.0 (REPXOS v3.0). ABCD está compuesto por varios módulos; por ejemplo, el módulo Adquisición, al cual se hace referencia en la presente investigación. REPXOS gestiona documentos en formato digital y los almacena en formato Dublin Core asociados al mismo. El bibliotecario almacena estos documentos de forma manual en el módulo Adquisición, lo que trae como consecuencia que exista duplicidad de la información y que en ocasiones los metadatos no estén correctos o falte alguno de ellos. El objetivo de la investigación, es implementar un componente en el módulo Adquisición del Sistema ABCD v3.0 que permita recolectar metadatos desde REPXOS v3.0 haciendo uso de las buenas prácticas que posee el protocolo OAI-PMH.

Palabras claves: adquisición, interoperabilidad, metadatos, Dublin Core, OAI-PMH, SIGB.

Índice de Contenido

Introducción	1
Capítulo 1: Fundamentación teórica para el componente del módulo Adquisición del Sistema ABCD v3.0	6
1.1 Introducción	6
1.2 Conceptos asociados al dominio del problema	6
1.2.1 Archivo	6
1.2.2 Protocolo	6
1.2.3 Recolección	6
1.2.4 Proveedor	6
1.2.5 Metadato	6
1.2.6 Registro bibliográfico	7
1.2.7 Interoperabilidad	7
1.3 Estudio de los protocolos de transferencia y recuperación de información	7
1.3.1 Protocolo OAI-PMH	7
1.3.2 Protocolo Z39.50	10
1.3.3 Conclusiones del estudio de los protocolos	11
1.4 Sistema para la Automatización de Bibliotecas y Centros de Documentación	14
1.5 Sistema Informático J-ISIS	16
1.5.1 Funciones de J-ISIS	16
1.5.2 Acceso a los registros bibliográficos almacenados en las bases de datos de J-ISIS	16
1.5.3 Estructura de los registros bibliográficos almacenados en J-ISIS	17
1.6 Estudio y análisis de los metadatos	17
1.6.1 Formato de metadatos Dublin Core	20
1.6.2 Etiquetas del módulo Adquisición	25
1.6.3 Equivalencia entre el formato de metadatos Dublin Core y los campos del módulo Adquisición	28
1.7 Metodología de desarrollo de software AUP-UCI	28

1.8 Herramientas y tecnologías utilizadas	30
1.8.1 Visual Paradigm versión 8.0	30
1.8.2 Eclipse versión Kepler Release 4.3	31
1.8.3 Gestores de Bases de Datos	31
1.8.4 Eclipse Virgo versión 3.6.3	32
1.8.5 RAP versión 3.0M5.....	33
1.8.6 OSGi Equinox versión 3.8.2.....	34
1.8.7 Spring Dynamic Modules versión 3.1.0.....	35
1.8.8 EclipseLink versión 2.3.....	36
1.9 Conclusiones del capítulo	36
Capítulo 2: Análisis y diseño de la propuesta de solución para el componente del módulo Adquisición del Sistema ABCD v3.0	37
2.1 Introducción	37
2.2 Descripción de la propuesta de la solución	37
2.3 Modelo de dominio.....	40
2.4 Especificación de los requisitos del software.....	40
2.4.1 Requerimientos funcionales.....	41
2.4.2 Técnicas de validación de requisitos	42
2.4.3 Requerimientos no funcionales.....	42
2.4.4 Listado de casos de uso del sistema	43
2.5 Modelo de casos de uso del sistema	43
2.5.2 Patrón de casos de uso	44
2.5.3 Actores del sistema	45
2.5.4 Matriz de trazabilidad.....	45
2.5.5 Descripción de los casos de uso	46
2.6 Descripción de la arquitectura.....	52

2.6.2 Estilo arquitectónico	52
2.6.3 Patrones de diseño GRASP	54
2.6.4 Patrones de diseño GoF	55
2.7 Modelo de diseño	55
2.7.2 Diagrama de clases del diseño	56
2.8 Conclusiones del capítulo	58
Capítulo 3: Implementación y pruebas del componente del módulo Adquisición del Sistema ABCD v3.0	59
3.1 Introducción	59
3.2 Implementación	59
3.3 Diagrama de despliegue	59
3.4 Pruebas de software	61
3.4.1 Estrategia de pruebas	61
3.4.2 Niveles de pruebas	61
3.4.3 Tipos de pruebas	62
3.4.4 Método de pruebas	62
3.4.5 Casos de pruebas	63
3.5 Conclusiones del capítulo	76
Conclusiones generales	77
Recomendaciones	78
Referencias bibliográficas	79
Bibliografía	83
Glosario de términos del dominio	84
Anexos	85
Anexo 1	85
Anexo 2	86
Anexo 3	88

Anexo 489

Índice de Tablas

Tabla 1 Sistemas bibliotecarios que utilizan el protocolo OAI-PMH.	10
Tabla 2 Sistemas bibliotecarios que utilizan el protocolo Z39.50.....	11
Tabla 3 Protocolos bajo las 4 libertades de software libre.....	13
Tabla 4 Tipología de los metadatos.	19
Tabla 5 Etiquetas de Dublin Core Cualificado.	25
Tabla 6 Etiquetas del módulo Adquisición.....	27
Tabla 7 Equivalencia de etiquetas entre Dublin Core y el módulo Adquisición.....	28
Tabla 8 Descripción de las fases de la metodología AUP-UCI.	30
Tabla 9 Lista de verbos OAI-PMH.....	39
Tabla 10 Actores del sistema.	45
Tabla 11 Descripción del caso de uso Gestionar proveedores.....	50
Tabla 12 Descripción del caso de uso Recolectar metadatos por colección.....	52
Tabla 13 Caso de Prueba Gestionar proveedores, SC 1 Adicionar proveedor.	68
Tabla 14 Caso de Prueba Gestionar proveedores, SC 2 Eliminar proveedor.	69
Tabla 15 Caso de Prueba Gestionar proveedores, SC 3 Modificar proveedor.	73
Tabla 16 Caso de Prueba Gestionar proveedores, SC 4 Listar proveedor.	74
Tabla 17 Descripción de las variables.	75
Tabla 18 Resultados de las pruebas de Caja Negra aplicadas al componente.	75
Tabla 19 Contenido del recurso Dublin Core Simple.....	85
Tabla 20 Propiedad intelectual Dublin Core Simple.	86
Tabla 21 Instancia Dublin Core Simple.	86
Tabla 22 Caso de Prueba Recolectar metadatos por colección.	93
Tabla 23 Caso de prueba Mostrar colecciones de un repositorio.	94
Tabla 24 Caso de Prueba Obtener información del proveedor de datos.	¡Error! Marcador no definido.

Tabla 25 Caso de Prueba Comprobar proveedor de datos.96

Índice de Ilustraciones

Ilustración 1 Estructura del protocolo OAI-PMH.	9
Ilustración 2 Número de implementaciones de los protocolos OAI-PMH y Z39.50.	14
Ilustración 3 Estructura de un registro bibliográfico.	17
Ilustración 4 Funcionamiento del protocolo OAI-PMH.	38
Ilustración 5 Diagrama de clases del modelo de dominio.	40
Ilustración 6 Diagrama de casos de uso del sistema.	43
Ilustración 7 Diagrama de caso de uso Gestionar proveedores antes de aplicar el CRUD completo.	44
Ilustración 8 Diagrama de casos de uso Gestionar proveedores después de aplicar el CRUD completo.	45
Ilustración 9 Matriz de trazabilidad Req_CU.	46
Ilustración 10 Arquitectura OSGI.	53
Ilustración 11 Diagrama de clases del diseño CU_Gestionar proveedores.	56
Ilustración 12 Clase ServicioAdquisicion.	57
Ilustración 13 Clase DAOBASE.	57
Ilustración 14 Clase DAOAdquisicion.	57
Ilustración 15 Clase Record.	58
Ilustración 16 Diagrama de despliegue.	60
Ilustración 17 Matriz de trazabilidad MC_Req.	87
Ilustración 18 Matriz de trazabilidad Req_Req.	87
Ilustración 19 Matriz de trazabilidad CU_CP.	88
Ilustración 20 Acta de aceptación.	89

Introducción

El desarrollo del proceso docente educativo en el que estamos inmersos ha sido influenciado extraordinariamente por los adelantos tecnológicos presentes en diversos sectores, específicamente en el de la informática. Es comprensible cómo el advenimiento de las tecnologías de la información y las comunicaciones han propiciado que las bibliotecas incorporen y hagan uso de herramientas que satisfagan las necesidades de los usuarios de adquirir conocimientos e información, así como el perfeccionamiento de los procesos de gestión bibliotecaria y los procesos de gestión de la información en las instituciones.

Las bibliotecas, desde la década de los noventa, son conscientes de que la información que deben ofrecer al usuario, se encuentra no solo en su propia institución sino también, en la de otros centros de documentación. El universo de información que hoy en día deben manejar, organizar y gestionar es amplio y muy diverso, al igual que la tecnología que tienen que utilizar. No solo es suficiente implantar un Sistema Integrado de Gestión de Bibliotecas (SIGB) sino que es necesario dar acceso a múltiples y variados documentos que proceden de fuentes diversas y que a veces, traen consigo una tecnología propia (Picco y Ortiz, 2012).

Las revistas electrónicas, los repositorios institucionales, los documentos en la red, las redes sociales, los libros electrónicos, los libros tradicionales en papel, las imágenes, los sonidos y una gran cantidad de recursos bibliográficos¹ que se amplía llegando a la nube (*Cloud computing*) han propiciado nuevas herramientas de búsqueda y acceso a la información: servidores de enlaces y metabuscadores que permiten al usuario un mejor y fácil acceso, además de los nuevos servicios que comienzan a implantarse: información directa al móvil, préstamo de lectores y de libros electrónicos (Picco y Ortiz, 2012).

Los catálogos bibliotecarios existen y siguen siendo necesarios, aunque hayan pasado de ser un elemento central a un servicio, entre otros, para acceder a la información. A pesar de que deban competir con servicios comerciales y gratuitos cada vez más potentes y eficaces que proyectan una sombra grande sobre ellos, siguen siendo de vital importancia, por lo que la comunidad bibliotecaria demanda la necesidad de nuevas herramientas para su creación, difusión, búsqueda e intercambio de información (Picco y Ortiz, 2012).

El proceso de intercambio de información bibliográfica es conocido actualmente como interoperabilidad, lo cual, para comunicarse y compartir datos, documentos, entre otros, de forma efectiva es relevante, pues posibilita tanto la interconexión libre entre uno o varios sistemas como la comunicación entre ellos. El avance tecnológico de la sociedad depende en gran medida de los procesos que los centros educacionales y sus bibliotecas sean capaces de informatizar y de la forma en la que puedan interactuar con otros SIGB.

¹Colección de objetos que aloja una biblioteca digital (también llamados ítems o materiales) entre los que se pueden considerar a los libros, periódicos, documentos (por ejemplo, páginas en formato HTML), así como diversos objetos multimedia (pinturas o imágenes, cintas o archivos de vídeo) (Gómez, 2012).

El uso de los sistemas integrados ha marcado un hito en la informatización de las bibliotecas y centros de documentación en todo el mundo, y por consiguiente, el modelo actual de la Educación Superior cubana requiere un uso intensivo de las Tecnologías de la Información y las Comunicaciones (TIC), las cuales, si de bibliotecas se trata, agrupan los elementos y las técnicas usadas en el tratamiento y la transmisión de la información, principalmente la informática, Internet y las telecomunicaciones. Siendo así de suma importancia que las bibliotecas universitarias del país estén a la altura del momento tecnológico que atraviesa nuestro proceso docente educativo (Borrero, 2013).

Este es el principal motivo por el cual las bibliotecas requieren algún sistema informático con el que puedan integrar todas sus áreas y que a su vez le permita la sistematización de su trabajo diario, con el objetivo de simplificar las tareas del centro, especialmente el proceso de catalogación, además de unificar en el catálogo público en línea (OPAC) todos sus materiales bibliográficos impresos y digitales y apoyar la labor de desarrollo de las colecciones y proceso de adquisiciones, para brindar un servicio eficiente a los usuarios (Borrero, 2013).

Dicho sistema debe brindar funcionalidades que permitan cooperar con otras bibliotecas para conformar un catálogo en línea de colaboración e intercambio de recursos y con el que se pueda contribuir en proyectos internacionales con otras bibliotecas de mayor dimensión. La solución más factible para resolver estas necesidades es el uso de los SIGB (Borrero, 2013).

Un Sistema Integrado de Gestión de Bibliotecas es una herramienta tecnológica que permite automatizar las operaciones bibliotecarias más comunes. Típicamente abarca la catalogación, circulación, consulta y adquisición de materiales. La tendencia de los mismos consiste en ampliar su oferta con nuevos productos o módulos integrados en sus sistemas de automatización (Martín, 2008).

La adquisición es la forma en la que la biblioteca incorpora a su colección obras previamente seleccionadas. Supone el abastecimiento de la colección bibliográfica, que constituye una de las actividades más importantes de la biblioteca, puesto que está dirigida a cubrir las necesidades de información de los usuarios en tiempo, forma y calidad. La adquisición es considerada un proceso bibliotecario fundamental y, por lo tanto, gestionada y/o dirigida por el bibliotecario (Gavilán, 2008).

Un ejemplo de SIGB es el Sistema ABCD (Sistema para la Automatización de Bibliotecas y Centros de Documentación), el cual es adaptable a disímiles bibliotecas debido al conjunto de características que posee, permite la gestión y administración de bases de datos, usuarios, estadísticas, adquisiciones y

préstamos, control de publicaciones periódicas y funciones de búsqueda (Borrero, 2013).

En la Universidad de las Ciencias Informáticas (UCI) existe el Centro de Informatización de la Gestión Documental (CIGED), dedicado al desarrollo de sistemas y servicios informáticos integrales de alta calidad y competitividad en la informatización y a la mejora de los procesos de gestión documental. Dicho centro cuenta con varios proyectos, entre ellos ABCD v3.0. Actualmente, la información bibliográfica que se gestiona en este sistema se encuentra almacenada en el sistema informático J-ISIS.

CIGED cuenta además con otro producto dentro de su catálogo, que junto a ABCD v3.0 tiene objetivos y fines que provoca que en ocasiones sean desplegados en una misma institución. Tal es el caso de REPXOS v3.0 (Sistema para Repositorios y Bibliotecas Digitales). El Sistema REPXOS v3.0 gestiona documentos en formato digital, almacenando el documento digital y metadatos en formato Dublin Core asociados al mismo. El bibliotecario almacena estos documentos de forma manual en el módulo Adquisición del Sistema ABCD v3.0, los cuales pasan a catalogación como documentos sin catalogar. Esto significa que el bibliotecario almacena toda la documentación registro por registro ingresando los datos en el sistema, trayendo como consecuencia una demora significativa en el proceso, lo que unido al margen de error humano hace que este proceso sea difícil y lento, posibilita la duplicidad de la información, puesto que se realiza el mismo proceso doblemente y puede que en ocasiones los metadatos no estén correctos o falte alguno de ellos.

La información en forma de metadatos también debe almacenarse en ABCD v3.0 mediante el proceso de catalogación, ya que muchos documentos digitales, tienen una copia física útil para los servicios que brinda la biblioteca, entre ellos el préstamo, lo cual permitirá que los usuarios puedan hacer búsquedas y obtener la copia en formato duro de los materiales digitales para dicho préstamo. Por tal motivo se necesita describir en el sistema ABCD v3.0 los registros bibliográficos de toda nueva adquisición que ingrese en la biblioteca tanto en forma física como electrónica utilizando el formato MARC21.

A partir de la situación problemática planteada se tiene el siguiente **problema a resolver**: ¿cómo contribuir al proceso de adquisición de los registros bibliográficos contenidos con el Sistema REPXOS v3.0 en el Sistema ABCD v3.0?

Se define como **objeto de estudio** los protocolos de intercambio de información bibliográfica y se enmarca como **campo de acción** el proceso de interoperabilidad del Sistema ABCD v3.0 con el sistema REPXOS v3.0.

Para la solución del problema se planteó como **objetivo general**: desarrollar un componente en el módulo Adquisición del Sistema ABCD v3.0 que permita la incorporación de registros bibliográficos desde el Sistema REPXOS v3.0 para el proceso de adquisición.

Para el desglose del objetivo general se definieron los siguientes **objetivos específicos**:

1. Describir el marco teórico referencial sobre los protocolos de intercambio de información bibliográfica.
2. Analizar el proceso de interoperabilidad del Sistema ABCD v3.0 con el Sistema REPXOS v3.0.
3. Diseñar mediante el uso de artefactos ingenieriles el componente para Adquisición a través del protocolo OAI-PMH en el Sistema ABCD v3.0.
4. Desarrollar el componente para Adquisición a través del protocolo OAI-PMH en el Sistema ABCD v3.0.
5. Validar mediante las pruebas de caja negra el componente para Adquisición a través del protocolo OAI-PMH en el Sistema ABCD v3.0.

Para el desarrollo de los objetivos específicos se utilizaron los siguientes métodos científicos:

Métodos teóricos:

Análisis Histórico-Lógico: facilitó el estudio y análisis de la evolución, desarrollo y funcionamiento de las bibliotecas, los centros de documentación y los SIGB, lo que permitió adecuar los datos necesarios a las especificaciones de la investigación para fundamentar la propuesta de solución a la problemática planteada.

Análítico-Sintético: permitió la extracción de los conceptos fundamentales relacionados con la gestión documental a lo largo de la realización del estudio del arte de la investigación y la recopilación de información necesaria a través de la revisión de documentos y artículos, además del análisis de las diferentes herramientas, tecnologías y metodología de desarrollo de software a emplear en la implementación del componente.

Modelación: permitió la representación, mediante el uso de diagramas, de las características del componente a desarrollar en el módulo Adquisición, así como el diseño de la interfaz del mismo, interacción, componentes y despliegue, además de la comprensión de las relaciones entre objetos y de la arquitectura

de la propuesta de solución.

Método empírico:

Observación: fue guía del estudio del estado del arte, permitiendo realizar un análisis objetivo de las relaciones esenciales y las características fundamentales de los principales sistemas que actualmente, pueden interoperar entre ellos.

El contenido del presente documento está compuesto por una introducción, tres capítulos de contenido, conclusiones generales, referencias bibliográficas, bibliografía y anexos.

Los capítulos están estructurados de la siguiente manera:

Capítulo 1: “Fundamentación teórica para el componente del módulo Adquisición del Sistema ABCD v3.0”: se describen los conceptos y criterios fundamentales que son de relevancia para la investigación, se realiza un estudio de varios sistemas de gestión bibliotecaria utilizados a nivel mundial y se exponen la metodología, las tecnologías a utilizar en la implementación del componente del módulo Adquisición, así como la metodología de desarrollo de software a emplear.

Capítulo 2: “Propuesta de solución para el componente del módulo Adquisición del Sistema ABCD v3.0”: se realiza una descripción detallada de la propuesta de solución. Se representan mediante un modelo de dominio los principales conceptos que se manejan en el contexto del sistema. Se realiza una descripción de los requisitos funcionales y no funcionales. Se elaboran las matrices de trazabilidad acordes a la metodología. Se confecciona el diagrama de casos de uso a partir de los requisitos obtenidos y se definen los casos de uso y los actores que se relacionan con cada uno de ellos.

Capítulo 3: “Implementación y prueba del componente del módulo Adquisición del Sistema ABCD v3.0”: se representan los elementos físicos necesarios para un correcto despliegue de la aplicación, empleando para ello el diagrama de despliegue. Se muestran componentes de la implementación. Se realiza la validación y prueba de la solución de acuerdo a los requisitos que debe cumplir para garantizar una calidad óptima, utilizando para ello las pruebas funcionales.

Capítulo 1: Fundamentación teórica para el componente del módulo Adquisición del Sistema ABCD v3.0

1.1 Introducción

El gran cúmulo de información que requiere toda investigación científica es de relevante importancia para sustentar las soluciones propuestas en el marco teórico. En el presente capítulo se muestran los conceptos asociados a la investigación. Se realiza un estudio de los protocolos de transferencia y recuperación de información. Se hace un resumen de la metodología, lenguajes, herramientas y tecnologías escogidas por el grupo de arquitectura de software de ABCD versión 3.0 para llevar a cabo la implementación del componente y se justifica la selección de los mismos.

1.2 Conceptos asociados al dominio del problema

Para lograr una mejor comprensión de todo el proceso investigativo se evidencian varios conceptos fundamentales que rigieron el mismo.

1.2.1 Archivo

Repositorio de documentos académicos. En un sentido más amplio la OAI² define el término “archivo” como un repositorio de información almacenada (Carpentier, 2003).

1.2.2 Protocolo

Un protocolo es un conjunto de normas que definen la comunicación entre sistemas. FTP (Protocolo de Transferencia de Ficheros) y HTTP (Protocolo de Transferencia de Hipertexto) son ejemplos de otros protocolos utilizados para la comunicación entre sistemas a través de Internet (Carpentier, 2003).

1.2.3 Recolección

En relación con la OAI, la recolección se refiere específicamente a la recogida de los metadatos de una serie de repositorios distribuidos en un almacén de datos combinados (Carpentier, 2003).

1.2.4 Proveedor

Persona o empresa que provee o abastece de todo lo necesario de insumos o servicios informáticos para un grupo, asociaciones o comunidades.

1.2.5 Metadato

Información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato u objeto

² Open Archive Initiative. (Iniciativa de Archivos Abiertos). Desarrolla y promueve normas de interoperabilidad que tienen por objetivo facilitar la difusión eficaz de los contenidos (Carpentier, 2003).

que tiene la finalidad de facilitar su recuperación, autenticación, evaluación, preservación e interoperabilidad (Rojas, 2010).

1.2.6 Registro bibliográfico

Conjunto de elementos informativos, organizados conforme a unas normas, que permiten identificar a una unidad documental de manera unívoca en vistas a su localización y posterior recuperación (Ríos, 2003).

1.2.7 Interoperabilidad

La interoperabilidad es *“la capacidad de un sistema de información de comunicarse y compartir datos, información, documentos y objetos digitales de forma efectiva (con una mínima o nula pérdida de su valor y funcionalidad), con uno o varios sistemas de información (siendo generalmente estos sistemas completamente heterogéneos, distribuidos y geográficamente distantes), mediante una interconexión libre, automática y transparente, sin dejar de utilizar en ningún momento la interfaz del sistema propio”* (Salinas, 2010).

1.3 Estudio de los protocolos de transferencia y recuperación de información

La transferencia y recuperación de la información en los SIGB, repositorios institucionales, entre otros, así como las herramientas de interconexión existentes entre ellos constituye una necesidad en los actuales sistemas de gestión documental. Es por ello que la creación, evolución y utilización de estándares de interoperabilidad juegan un papel importante en la búsqueda e intercambio de información bibliográfica. Para una mejor comprensión del presente estudio se reflejan las características fundamentales de dichos estándares.

1.3.1 Protocolo OAI-PMH

El protocolo OAI-PMH (Open Archives Initiative - Protocol Metadata Harvesting, por sus siglas en inglés o Iniciativa Abierta de Archivos – Protocolo de Recolección de Metadatos, por sus siglas en español) es una herramienta de interoperabilidad que posibilita el intercambio de metadatos sobre cualquier material almacenado en soporte electrónico. Esta transferencia se realiza desde los proveedores de servicio a través de búsquedas hacia los proveedores de datos (repositorios de archivos) (Salinas, 2010).

Historia del surgimiento del protocolo OAI-PMH

En octubre de 1999 se organizó una reunión en Santa Fe (Nuevo México, USA), donde los participantes eran especialistas en bibliotecas digitales, con la idea de que la interoperabilidad de archivos de EPrints era clave para aumentar su impacto entre la comunidad académica (Barrueco, 2008).

El resultado de la reunión fue un conjunto de acuerdos técnicos y organizativos conocidos como la Convención de Santa Fe (Barrueco, 2008).

Las especificaciones revisadas fueron hechas públicas en enero de 2001 con la publicación del Open Archives Initiative – Protocol for Metadata Harvesting (OAIPMH) versión 1.0. La intención era que este protocolo, con mínimas modificaciones, permaneciera estable al menos durante un año, mientras las distintas comunidades lo probaban y experimentaban con él (Barrueco, 2008).

Inmediatamente después de la difusión de la versión 1 comenzó el trabajo del comité técnico para tratar los problemas de definición o funcionalidad que se fueran descubriendo. Ese trabajo desembocó en la elaboración de la versión 2 del protocolo anunciada en junio de 2002. Los principales cambios que se introdujeron fueron relacionados con la clarificación de ambigüedades o mejores medios para expresar las funcionalidades existentes. Es decir, no se introdujeron cambios sustanciales (Barrueco, 2008).

Estructura del protocolo OAI-PMH

En la estructura del protocolo OAI-PMH intervienen dos factores fundamentales: el Proveedor de Datos (DP, *Data Provider*, por sus siglas en inglés) y el Proveedor de Servicios (SP, *Service Provider*, por sus siglas en inglés).

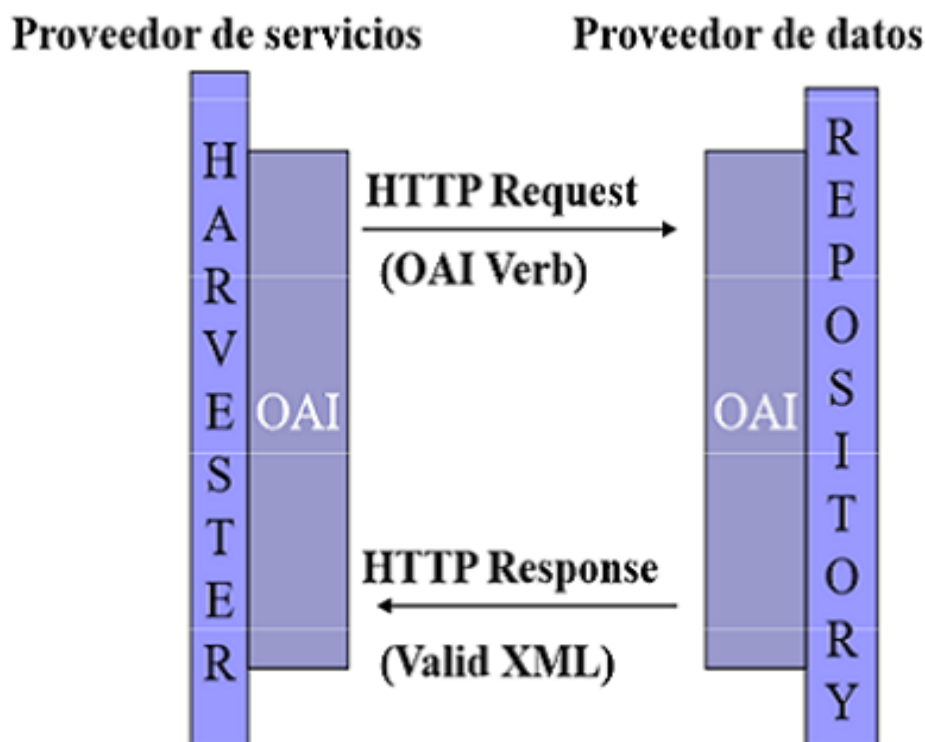


Ilustración 1 Estructura del protocolo OAI-PMH³.

Esta estructura de interoperabilidad demuestra el enfoque de harvesting (recopilación), en la que la información es transmitida desde la fuente remota (servidor) al destino (cliente) donde a la información se le añadirá un valor agregado.

Un Proveedor de Datos mantiene uno o varios repositorios (servidores web) que soportan OAI-PMH como medio de exponer sus metadatos (Carpentier, 2003).

Un proveedor de servicios envía peticiones OAI-PMH a los proveedores de datos y utiliza los metadatos como base para la creación de servicios de valor añadido. De esta forma un Proveedor de Servicios "recolecta" los metadatos expuestos por los Proveedores de Datos (Carpentier, 2003).

Sistemas bibliotecarios que utilizan el protocolo OAI-PMH

³ Extraído de <http://102novadoc.es/masinfo/oai-deusto.pdf>.

Existen numerosas instituciones bibliotecarias que hacen uso del estándar OAI-PMH para organizar, manejar y difundir su información bibliográfica. A continuación, en la tabla 1 se evidencian varios ejemplares:

Sistemas bibliotecarios	Dirección disponible
E-Prints Complutense	http://eprints.ucm.es/cgi/oai2
ABACUS. Repositorio de Producción Científica	http://abacus.universidadeuropea.es/oai/request
ABC Journal of Advanced Research	http://journals.abc.us.org/index.php/abcjar/oai
Aesthetic Investigations	http://www.aestheticinvestigations.eu/index.php/journal/oai
ARCHIVE OF EUROPEAN INTEGRATION	http://aei.pitt.edu/perl/oai2

Tabla 1 Sistemas bibliotecarios que utilizan el protocolo OAI-PMH⁴.

Fuente: Elaboración propia

1.3.2 Protocolo Z39.50

Z39.50 (Information Retrieval); Application Service Definition and Protocol Specification. ANSI/NISO Z39.50-2003) es un protocolo para la recuperación de información basado en la estructura cliente/servidor que facilita la interconexión de sistemas informáticos. El objetivo principal de este estándar consiste en permitir al usuario realizar búsquedas en bases de datos que cuenten con un servidor Z39.50, sin tener que conocer para ello las sintaxis de búsqueda que utilizan dichos sistemas (Salinas, 2010).

Historia del surgimiento del protocolo Z39.50

En 1988, el Instituto Nacional Americano de Estándares (ANSI) aprobó la norma Z39.50 versión 1. No obstante, no fue hasta la publicación de su versión 2, en 1992, cuando la comunidad bibliotecaria comenzó a interesarse por su aplicación. En los años siguientes se continuó trabajando en el desarrollo del estándar y se contó con la colaboración de otros países, como Canadá, Australia y naciones europeas. Esto dio lugar a la aprobación de su versión 3 en 1995, compatible con la versión 2, pero mejorada hasta tal punto que, en 1998, la Organización Internacional para la Normalización (ISO) decidió adoptar la norma americana Z39.50 como estándar internacional bajo el nombre ISO 23950-1998. Además de adquirir una dimensión verdaderamente internacional, su aceptación como norma ISO consolidó el protocolo TCP/IP de Internet

⁴ Para mayor información, estos listados se encuentran disponibles en la dirección <https://www.openarchives.org/pmh>.

como su medio de aplicación (Biblion, 2010).

Estructura del protocolo Z39.50

Z39.50 se basa en una estructura cliente-servidor donde un cliente u origen también conocido como Cliente Z es el que se comunica con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y seguridad. El usuario interactúa con el cliente de forma gráfica. El servidor o target conocido como Servidor Z, es quien proporciona un servicio al cliente y entrega los resultados de una tarea específica, estableciendo procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Emplea además, una estructura de datos, reglas de intercambio y un transporte de información confiable como son TCP/IP (Transmission Control Protocol/Internet Protocol) y OSI (Open System Interconnection) (Salinas, 2010).

Sistemas bibliotecarios que utilizan el protocolo Z39.50

A continuación, se evidencian varios ejemplos de sistemas bibliotecarios que hacen uso del protocolo Z39.50:

Sistema bibliotecarios	Dirección disponible
Library of Congress	https://www.loc.gov/
Abilene Library Consortium	https://www.alc.org
Murray State University	https://www.murraystate.edu
Albany State University	https://www.asurams.edu
Cardiff University	https://www.cardiff.ac.uk

Tabla 2 Sistemas bibliotecarios que utilizan el protocolo Z39.50⁵.

Fuente: Elaboración propia

1.3.3 Conclusiones del estudio de los protocolos

Con el estudio de los protocolos de transferencia y recuperación de información, se adquirió conocimientos

⁵ Para mayor información, estos listados se encuentran disponibles en la dirección <http://www.loc.gov/z3950/gateway.html>.

sobre sus características estructurales y funcionales. Como complemento de este estudio se indagó en los sistemas bibliotecarios que hacen uso de estos protocolos para interoperar con otros sistemas.

Luego de haber analizado estos estándares de interoperabilidad fue preciso seleccionar uno de ellos para que el Sistema ABCD interopere con otros sistemas, teniendo en cuenta dos criterios:

- Software libre como iniciativa para la Universidad de las Ciencias Informáticas.
- Implementación a nivel mundial.

Software libre como iniciativa para la Universidad de las Ciencias Informáticas

Según la Fundación para Software Libre (FSF, *Free Software Foundation*, por sus siglas en inglés), del proyecto GNU, software libre es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software⁶.

Para que un software determinado sea libre, el usuario debe poseer las cuatro libertades esenciales (FSF, 2017):

- Libertad 0: La libertad de ejecutar el programa como se desea, con cualquier propósito.
- Libertad 1: La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera. El acceso al código fuente es una condición necesaria para ello.
- Libertad 2: La libertad de redistribuir copias para ayudar a su prójimo.
- Libertad 3: La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

Este es un criterio muy importante para aquellas entidades productivas que requieran la utilización de manera libre del software. En la Resolución 179/14 perteneciente a la Dirección de Servicios Jurídicos de la UCI se plantea en el acuerdo No. 084 del año 2004 que “el Comité Ejecutivo del Consejo de Ministros identificó la necesidad de ejecutar acciones que garantizaran la migración ordenada y progresiva hacia aplicaciones y plataformas de código abierto, en concordancia con el desarrollo del proceso de informatización de la sociedad como parte de la ejecución por el país de una política orientada a alcanzar la seguridad, invulnerabilidad e independencia tecnológica” (UCI, 2014). La tabla 4 ejemplifica las 4

⁶ <http://www.gnu.org/philosophy/free-sw.es.html>.

libertades de software libre con respecto a los estándares de interoperabilidad estudiados. Para ello se realizaron consultas en los sitios oficiales que los financian.

Libertades de software libre	Protocolos	
	OAI-PMH	Z39.50
0. La libertad de ejecutar el programa como se desea, con cualquier propósito.	Sí	Sí
1. La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera. El acceso al código fuente es una condición necesaria para ello.	Sí	Sí
2. La libertad de redistribuir copias para ayudar a su prójimo.	Sí	No
3. La libertad de distribuir copias de sus versiones modificadas a terceros.	Sí	Sí

Tabla 3 Protocolos bajo las 4 libertades de software libre.

Fuente: Elaboración propia

Estas características evidencian que el protocolo Z39.50 no cumple con la Libertad 2, pues la totalidad de sus implementaciones validadas por la Biblioteca del Congreso de los Estados Unidos son de código cerrado. En el caso específico del protocolo OAI-PMH las herramientas de código abierto implementadas por la OAI se encuentran disponibles en la dirección <https://www.openarchives.org/pmh/tools/>.

Implementación a nivel mundial de los protocolos estudiados

El porcentaje de implementaciones a nivel mundial de los protocolos de búsqueda e intercambio de información bibliográfica es otro de los factores estadísticos que tienen en cuenta los actuales sistemas bibliotecarios para su adopción. Para el caso de los protocolos estudiados, se puede estimar el número de implementaciones, ya que existen entidades encargadas de su desarrollo, mantenimiento e implementación. Estas instituciones recomiendan registrar dichas utilidades en sus sitios web oficiales. Aunque el protocolo Z39.50 es un software privativo, varias entidades lo usan para intercambiar información; sus implementaciones están registradas en el sitio web de la Biblioteca del Congreso de los Estados Unidos.

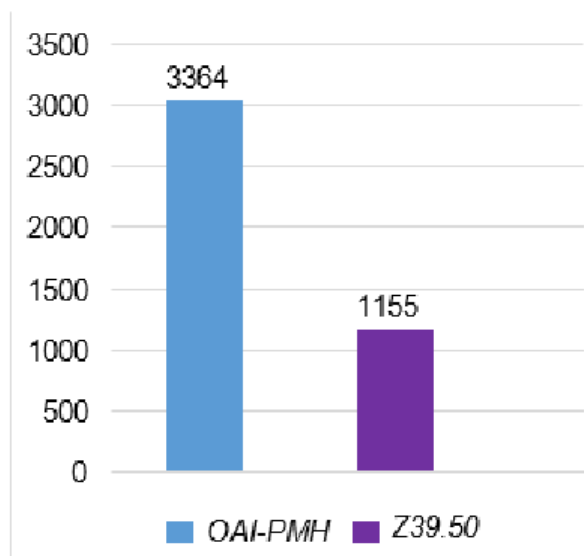


Ilustración 2 Número de implementaciones de los protocolos OAI-PMH y Z39.50⁷.

Fuente: Elaboración propia

La ilustración 2 muestra el número de implementaciones a nivel mundial de los protocolos previamente mencionados. El protocolo OAI-PMH, cuenta con 3364 implementaciones y el protocolo Z39.50 dispone de 1155. Con estas estadísticas numéricas se evidencia la enorme aceptación que posee el estándar OAI-PMH desde su fecha de creación hasta la actualidad.

Teniendo en cuenta el análisis realizado acerca de los criterios para la adopción de los protocolos de búsqueda e intercambio de información bibliográfica, se selecciona el estándar OAI-PMH, puesto que cumple con las 4 libertades de software libre que propone la FSF y es el de mayor implementación a nivel mundial. Otro aspecto de importancia es que el proyecto ABCD utiliza este protocolo para transferir información entre repositorios.

1.4 Sistema para la Automatización de Bibliotecas y Centros de Documentación

ABCD es un sistema para la Automatización de Bibliotecas y Centros de Documentación, perteneciente al programa vllir-uos (RED DE COOPERACIÓN UNIVERSITARIA), el cual está enfocado al fortalecimiento del papel de las TIC en las universidades cubanas para el desarrollo de la sociedad. El mismo apoya las asociaciones entre universidades y colegios universitarios en Flandes (Bélgica) y países del Sur. ABCD inicio con la versión 1.2 pero al ser analizada por los especialistas del centro CIGED de la UCI en

⁷ Las implementaciones del protocolo z39.50 se encuentran disponible en la dirección <http://www.loc.gov/z3950>. Las implementaciones del protocolo OAI-PMH se encuentran registradas en el sitio Web de la OAI disponible en la dirección <https://www.openarchives.org/Register/BrowseSites>.

conjunto con doctores y especialistas en Ciencia de la Información de las universidades miembros de la Red se detectaron ciertas deficiencias del sistema que concertaron la necesidad de implementar una versión superior, para de esta forma garantizar una mejor gestión de los procesos bibliotecarios involucrados. A partir de esta necesidad, se desarrolló ABCD v3.0.

ABCD v3.0 cuenta con 6 módulos con los que puede automatizar la información requerida:

Módulo de Catalogación: Permite la construcción del instrumento que facilita el acceso a los documentos: el catálogo. Es esencial en cualquier SIGB porque sin un catálogo detrás estos sistemas no podrían funcionar. Aquí se crean registros bibliográficos, se normalizan entradas y se mantienen los índices.

Módulo de Circulación: Establece relaciones temporales entre la información bibliográfica y los registros de usuarios y sirve fundamentalmente para desarrollar las operaciones préstamo a domicilio. Gracias a la parametrización de la política de circulación de la biblioteca es posible gestionar una gran cantidad de operaciones relacionadas con la circulación física de los documentos que posee la biblioteca: permite gestionar los usuarios, la devolución, las reservas, el envío de avisos, las sanciones por retrasos en devolución, hacer recuentos de uso en sala, entre otros.

Módulo de Adquisición: Ayuda a gestionar la tenencia de nuevos documentos en la Biblioteca. Utiliza información bibliográfica existente en el catálogo para ejecutar los pedidos, o permite la incorporación en el catálogo de descripciones suficientes para realizar el pedido. Posibilita la gestión de los proveedores y los fondos presupuestarios destinados a la adquisición, lanzar reclamaciones de pedidos no recibidos o cancelar peticiones, además, de gestionar desideratas.

Módulo OPAC: El Catálogo Público de Acceso en Línea (OPAC, Online Public Access Catalogue, por sus siglas en inglés) muestra públicamente el contenido del catálogo a través de Internet. Es decir, es la interfaz que permite a los usuarios acceder a los recursos bibliográficos del sistema e interactuar con ellos. Las últimas generaciones de OPAC son web, y suelen incorporar a las tradicionales operaciones de búsqueda y recuperación de la información, otros servicios de valor añadido, algunos de ellos personalizados.

Portal Web: Cuenta con un portal web para la interacción del usuario con el sistema.

Sistema para el almacenamiento de información: La información que gestiona el Sistema ABCD se encuentra almacenada en el sistema informático J-ISIS.

1.5 Sistema Informático J-ISIS

El sistema informático Java - Conjunto Integrado para Servicios de Información (J-ISIS, *Java - Integrated Set for Information Services*, por sus siglas en inglés) es un Sistema Gestor de Bases de Datos que permite la elaboración y administración de bases de datos. Fue creado por el francés Jean-Claude Dauphin, desarrollador de software de la UNESCO HQ⁸ en octubre del año 2008 (Dauphin, 2015).

Con el estudio de este sistema, se pretende obtener conocimiento acerca de sus funcionalidades. En la investigación, solo será de interés la función relacionada al acceso de los registros bibliográficos, ya que el contenido de información que almacenan es el que se necesita intercambiar con otros sistemas bibliotecarios.

1.5.1 Funciones de J-ISIS

El SGBD J-ISIS presenta las siguientes funciones (Dauphin, 2015):

- Crear bases de datos desde el inicio definiendo, de manera personal, su contenido (tablas, registros bibliográficos, entre otros).
- Acceder a las bases de datos almacenadas.
- Añadir de nuevos registros bibliográficos en una base de datos determinada.
- Acceder a los registros bibliográficos de las bases de datos almacenadas en el sistema.
- Modificar, corregir o suprimir registros bibliográficos existentes en una base de datos.
- Recuperar registros bibliográficos por sus contenidos a través de búsquedas avanzadas.
- Exportar los registros bibliográficos o porciones de ellos acorde a los requerimientos del usuario.
- Ordenar los registros bibliográficos en cualquier secuencia deseada.

1.5.2 Acceso a los registros bibliográficos almacenados en las bases de datos de J-ISIS

El acceso a los registros bibliográficos almacenados en el SGBD J-ISIS es a través del Número de Archivo Maestro (MFN, *Master File Number*, por sus siglas en inglés). El MFN es un conjunto de archivos denominados Berkeley DB⁹ en los cuales se almacenan todos los registros bibliográficos de una base de datos determinada albergada en el SGBD J-ISIS. Cada registro se identifica por un único MFN asignado

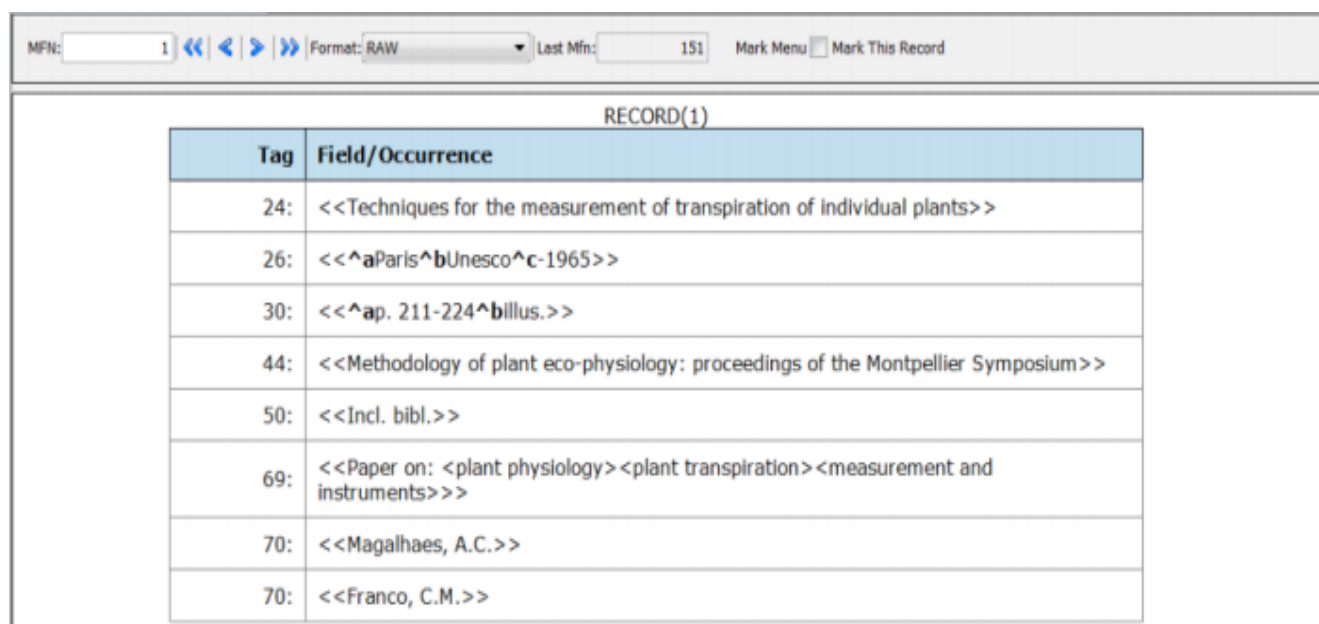
⁸ <http://www.unesco.org>.

⁹ https://www.usenix.org/legacy/event/usenix99/full_papers/olson/olson.pdf.

automáticamente por el SGBD J-ISIS (Dauphin, 2015).

1.5.3 Estructura de los registros bibliográficos almacenados en J-ISIS

La información que se almacena en las bases de datos del SGBD J-ISIS es tratada con el término de registro bibliográfico (*record*, en inglés). Se componen por conjuntos de etiquetas, campos u ocurrencias, tipo de formato de metadato y por un MFN. Los campos u ocurrencias (*field/occurrence*, en inglés) almacenan los datos en específico de un registro bibliográfico. Las etiquetas (*tag*, en inglés) son asignaciones numéricas a cada campo u ocurrencia de un registro bibliográfico respectivamente. Los tipos de formatos de metadatos (*format*, en inglés) son las distintas formas de catalogar un registro bibliográfico y el MFN es un identificador numérico único de cada registro bibliográfico que permite el acceso a estos para la extracción de sus datos. A continuación, se muestra un ejemplo de la estructura de un registro bibliográfico almacenado en el SGBD J-ISIS.



Tag	Field/Occurrence
24:	<<Techniques for the measurement of transpiration of individual plants>>
26:	<<^aParis^bUnesco^c-1965>>
30:	<<^ap. 211-224^billus.>>
44:	<<Methodology of plant eco-physiology: proceedings of the Montpellier Symposium>>
50:	<<Incl. bibl.>>
69:	<<Paper on: <plant physiology><plant transpiration><measurement and instruments>>>
70:	<<Magalhaes, A.C.>>
70:	<<Franco, C.M.>>

Ilustración 3 Estructura de un registro bibliográfico¹⁰.

1.6 Estudio y análisis de los metadatos

El medio por el cual se cataloga de un documento digital es a través de los metadatos. Son muy utilizados en el ambiente bibliotecario para describir documentos, objetos de información o de aprendizaje (tesis,

¹⁰ Extraído de un registro bibliográfico almacenado en el SGBD J-ISIS.

libros, videos, artículos u otro recurso que apoye la difusión del conocimiento) para ser visualizados sobre un ambiente web.

A continuación, se ejemplifican los beneficios que aporta la utilización de metadatos, específicamente para los usuarios que realizan búsquedas bibliográficas (Argueta, 2013).

- Encontrar los datos buscados.
- Conocer información que es clave en los datos.
- Comprender en profundidad la información.
- Localizar datos (dentro y fuera de la organización).
- Transferir e interpretar los datos correctamente.

Existen diversos tipos de metadatos que son utilizados en las bibliotecas digitales (Siza, 2010):

En la siguiente tabla se recogen los diferentes tipos de metadatos, así como una breve descripción de en qué consisten cada uno de ellos y ejemplos de su uso.

Tipología de los metadatos:

Tipos de Metadatos	Descripción	Ejemplos de uso
Administrativos	Metadatos usados para gestionar y administrar recursos de información.	<ul style="list-style-type: none"> • Información sobre adquisición. • Seguimiento de derechos. • Documentación de requerimientos de acceso legal. • Información sobre la posición. • Criterios de selección. • Control de versiones.

Descriptivos	Metadatos usados para describir o identificar recursos de información.	<ul style="list-style-type: none"> • Catálogos. • Índices especializados. • Relaciones entre recursos de información. • Anotaciones hechas por usuarios.
Preservación	Metadatos relacionados con la preservación de recursos de información.	<ul style="list-style-type: none"> • Documentación de condiciones físicas de los recursos de información. • Documentación de acciones llevadas a cabo para generar versiones.
Técnicos	Metadatos relacionados con el funcionamiento de los sistemas.	<ul style="list-style-type: none"> • Documentación sobre <i>hardware</i> y <i>software</i>. • Información sobre formatos, razones de comprensión, rutinas de escalado, entre otros. • <i>Password</i>, llaves de encriptación.
Uso	Metadatos relacionados con el nivel de y tipología de uso de los recursos de información.	<ul style="list-style-type: none"> • Muestra de registros. • Seguimiento de uso y de usuarios. • Reusado de contenido. • Información sobre versiones

Tabla 4 Tipología de los metadatos¹¹.

¹¹ Para mayor información este listado se encuentra disponible en la dirección <http://www.bdigital.unal.edu.co/1758/>.

Según estos tipos de metadatos, dentro de los sistemas bibliotecarios se hace uso de los metadatos descriptivos pues lo que se requiere almacenar son aquellos atributos que están relacionados intrínsecamente con el documento y que se pueden diferenciar unos de otros.

A continuación, se analizará el formato de metadatos descriptivos *Dublin Core* y los campos existentes en el módulo Adquisición, para establecer la relación entre ambos. *Dublin Core* es analizado porque el componente informático debe mostrar estos registros bibliográficos bajo las pautas estructurales que él establece.

1.6.1 Formato de metadatos Dublin Core

Dublin Core (DC) es un modelo de metadatos elaborado y auspiciado por la Iniciativa de Metadatos Dublin Core (DCMI, *Dublin Core Metadata Initiative*, por sus siglas en inglés), 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 y permitir sistemas más inteligentes del descubrimiento del recurso (Rojas, 2010).

El estándar de metadatos *Dublin Core* es un eficaz conjunto de elementos para describir una amplia gama de recursos de red. La norma del *Dublin Core* conlleva dos niveles: Simple y Cualificado. El *Dublin Core* Simple contiene quince elementos; el *Dublin Core* Cualificado contiene un elemento adicional, la audiencia, así como un grupo de elementos de matización (denominados por ello, cualificadores) que refinan la semántica de los elementos de tal forma que pueden ser útiles para la recuperación y localización de recursos en Internet (Rojas, 2010).

DC es uno de los modelos de metadatos más conocidos para catalogar documentos bajo un ambiente *Web* debido a la estructura representativa que posee. Este estándar se creó a partir de un taller de metadatos realizado precisamente en la ciudad de *Dublin, Ohio* (Estados Unidos) en 1995 (Pablo y Ramírez, 2010).

Cada uno de los elementos del conjunto básico de *Dublin Core* es opcional y puede repetirse.

El conjunto de elementos de metadatos *Dublin Core* es un estándar para la descripción de recursos (algo que tiene identidad) entre distintos dominios informativos (Rojas, 2010).

Del presente formato se estudia los tipos de normas que posee y las estructuras para catalogar información bibliográfica.

Normativas de *Dublin Core*

Dublin Core presenta sus normativas. En este caso, los formatos son dos y se clasifican en ***Dublin Core Simple*** (sin cualificar) y ***Dublin Core Cualificado***.

Dublin Core Simple

Dublin Core Simple presenta 15 etiquetas para catalogar los recursos bibliográficos: Título, Creador, Colaborador, Editor, Fecha de publicación, Identificador, Materia, Descripción, Cobertura, Tipo de recurso, Idioma, Formato, Fuente, Relación con otros documentos y Derechos. [\(Ver anexo 1\)](#)

Dublin Core Cualificado

La norma *Dublin Core Cualificado* posee la característica de especificar (cualificar) algunas de las etiquetas que presenta la norma *Dublin Core Simple*. Añade 4 etiquetas para catalogar: Método de incremento, Periodicidad del incremento, Audiencia y Procedencia.

En la siguiente tabla se muestran las etiquetas de *Dublin Core Cualificado*, una breve descripción de que significan cada una de ellas, así como el cualificador DC y la descripción de dicho cualificador.

Etiquetas de *Dublin Core Cualificado*

Etiquetas <i>DC</i>	Descripción	Cualificador <i>DC</i>	Descripción del Cualificador <i>DC</i>
<i>DC.Accrual Method</i>	Método de incremento. Es el método que se usa para agregar elementos a una colección.		
<i>DC. Accrual Periodicity</i>	Periodicidad del incremento. La frecuencia con el cual los elementos se agregan a la colección.		
<i>DC.Audence</i>	Audiencia. Clase de entidad para quién el recurso pretende ser útil. La clase de entidad puede ser determinada por el autor, editor o por		

	<p>una tercera parte. Guía para la creación del contenido: Los términos incluidos en Audiencia se utilizan preferentemente en el contexto de vocabularios controlados formales o informales. Ninguno de ellos, actualmente, ha sido registrado o recomendado por <i>DCMI</i>. En ausencia de vocabularios controlados recomendados, se anima a los implementadores a desarrollar listas locales de valores, y usarlos consistentemente.</p>		
<i>DC.Contributor</i>	<p>Colaborador. Responsable de hacer colaboraciones al contenido del recurso.</p>		
<i>DC.Coverage</i>	<p>Cobertura. Ámbito del contenido del recurso. Puede tratarse de una especificación geográfica, temporal o legal.</p>	<p><i>Spatial</i> <i>Spatial - ISO3166</i> <i>Spatial - TGN</i> <i>Temporal</i></p>	
<i>DC.Date</i>	<p>Fecha. Fecha asociada a la creación o modificación del recurso. Se suele seguir la notación AAAA-MM-DD (año, mes, día respectivamente).</p>	<p><i>Copyrighted</i> <i>Created</i> <i>Issued</i> <i>Modified</i> <i>Valid</i></p>	
<i>DC.Description</i>	<p>Descripción. Descripción del contenido del recurso. Puede incluir un resumen, una tabla de contenidos, entre otros.</p>	<p><i>Abstract</i> <i>Table of Contents</i></p>	

<i>DC.Format</i>	Formato. Descripción física del recurso, como su tamaño, duración, dimensiones, si son aplicables. Se suelen usar tipos <i>MIME</i> .	<i>Extent</i> <i>IMT</i> <i>Medium</i>	
<i>DC.Identifier</i>	Identificación. Referencia unívoca para el contenido del recurso. Por ejemplo, un <i>URI</i> (Identificador de Recurso Uniforme) o un <i>ISBN</i> (código normalizado internacional para libros).	<i>URI</i>	
<i>DC.Language</i>	Idioma. El idioma del contenido del recurso.	<i>ISO369-2</i> <i>ISO639-2</i> <i>ISO639-3</i> <i>RFC1766</i> <i>RFC3066</i> <i>RFC4646</i>	
<i>DC.Provenance</i>	Procedencia. Declaración de cualquier cambio en la propiedad y custodia del recurso desde su creación, que sea significativo para su autenticidad, integridad e interpretación.		
<i>DC.Relation</i>	Relación. Una referencia a un recurso relacionado con el contenido.	<i>HasFormat</i> <i>HasFormatURI</i> <i>HasPart</i> <i>HasPartURI</i> <i>HasVersion</i> <i>HasVersionURI</i>	

		<i>IsFormatOf</i> <i>IsFormatOfURI</i> <i>IsPartOf</i> <i>IsPartOfURI</i> <i>IsReferencedBy</i> <i>IsReplacedBy</i> <i>IsReplacedByURI</i> <i>IsVersionOf</i> <i>IsVersionOfURI</i> <i>Replaces</i> <i>ReplacesURI</i> <i>Requires</i>	
<i>DC.Rights</i>	Derechos. Información sobre los derechos de la propiedad intelectual del recurso, como por ejemplo el <i>copyright</i> .	<i>AccessRights</i>	
		<i>RightsHolder</i>	Titulares de los derechos. Persona u organización que posee o que gestiona los derechos sobre el recurso. La práctica más recomendada es usar la <i>URI</i> o el nombre del titular de los derechos.

<i>DC.Source</i>	Fuente. Referencia al recurso del que deriva el documento actual.	<i>URI</i>	
<i>DC.Subject</i>	Materias y palabras clave. El tema del contenido del recurso.	<i>DDC</i> <i>LCC</i> <i>LCSH</i> <i>MeSH</i> <i>NLM</i> <i>TGN</i> <i>UDC</i>	
<i>DC.Title</i>	Título. El nombre dado al recurso.	<i>Alternative</i>	
<i>DC.Type</i>	El tipo o categoría del contenido. Palabras clave de un vocabulario que describen la naturaleza del recurso.	<i>DCMI Type Vocabulary</i>	

Tabla 5 Etiquetas de Dublin Core Cualificado¹².

La siguiente tabla muestra las etiquetas del módulo Adquisición, el nombre de cada una de ellas y si es obligatorio o no.

1.6.2 Etiquetas del módulo Adquisición

Etiquetas	Nombre	Obligatorio
1	Número de control	Si
2	Título	Si
3	Autor	Si
4	Editor	No

¹² <https://www.loc.gov/marc/marc2dc.html#ldr06conversionrules>.

5	Año de publicación	No
6	Tipo de material	Si
7	Sala	Si
8	Autor corporativo	No
9	Ciudad	No
10	Código barra	No
11	Edición	No
12	Estado del material	No
13	Nota pública	No
14	País	No
15	Tomo	No
16	Volumen	No
17	Localización en estantería	No
18	Localización permanente	No
19	Restricciones de uso	No
20	Fecha de adquisición	No
21	Precio	Si
22	Moneda	No
23	Vía de adquisición	Si
24	Proveedor	No

25	Donante	No
26	Condiciones	No
27	Referencia	No
28	Cantidad recibida	No
29	Descripción del objeto	No
30	Tipo de objeto(nuevo/copia)	No
31	Institución	No
32	Canjeado por	No
33	Cantidad solicitada	No
34	Número de cotización	No
35	Control de Adquisición	No
36	Vencimiento de la suscripción	No
37	Prioridad de Adquisición	No
38	Notas administrativas	No
39	Históricos de compras	No

Tabla 6 Etiquetas del módulo Adquisición¹³.

Fuente: Elaboración propia

Luego de haber realizado el estudio de la estructura de los formatos de metadatos Dublin Core y los campos del módulo Adquisición para catalogar la información bibliográfica digital, se establece la equivalencia de etiquetas entre las normas Dublin Core Cualificado y el módulo Adquisición.

¹³ Extraído de la base de datos del módulo Adquisición del Sistema ABCD v3.0.

1.6.3 Equivalencia entre el formato de metadatos Dublin Core y los campos del módulo Adquisición

Con el estudio de las características del formato de metadatos Dublin Core y los campos del módulo Adquisición, se obtuvo conocimiento acerca de la catalogación de materiales bibliográficos digitales en los sistemas bibliotecarios. Para ello, se hace uso de etiquetas que los describen y ordenan. Adquisición contiene una serie de elementos estructurales (etiquetas) que tienen equivalencia con los de Dublin Core. Cada una de estos elementos contiene un número único para hacer referencia a la información que se está preservando en el campo u ocurrencia del registro bibliográfico.

A continuación, se evidencia la equivalencia entre las etiquetas de Dublin Core Cualificado y el módulo Adquisición.

Equivalencia de etiquetas entre Dublin Core y el módulo Adquisición

Etiquetas DC	Etiquetas módulo Adquisición
<i>DC.Title</i>	<i>Título</i>
<i>DC.Source</i>	<i>Referencia</i>
<i>DC.Creator</i>	<i>Autor</i>
<i>DC.Publisher</i>	<i>Editor</i>
<i>DC.Date</i>	<i>Año de publicación</i>
<i>DC.Type</i>	<i>Tipo de material</i>

Tabla 7 Equivalencia de etiquetas entre Dublin Core y el módulo Adquisición.

Fuente: Elaboración propia

1.7 Metodología de desarrollo de software AUP-UCI

La forma de planificar un proyecto y realizar las estimaciones del tiempo son factores determinantes en la culminación exitosa de todo desarrollo de software, por lo que es necesario utilizar una metodología que no aplique un único cronograma tipo y no fuerce el método de estimación definido por la UCI que responde en su gran mayoría a la metodología RUP (UCI, 2016).

El esfuerzo en tiempo y en personas de la variante de RUP es mucho mayor que la posibilidad de converger a los proyectos hacia una sola metodología de desarrollo. Por tanto, se decidió escoger una metodología

para ser adaptada a lo que ya la Universidad ha estado proponiendo como ciclo de vida de los proyectos, sin alejarse de lo que hasta el momento se ha trabajado e introducir la menor cantidad de cambios posibles (UCI, 2016).

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP, por sus siglas en inglés) es una versión simplificada del Proceso Unificado de Rational (RUP, por sus siglas en inglés). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP (UCI, 2016).

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI (UCI, 2016).

Descripción de las Fases

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, la cual se nombrará Ejecución y se agrega la fase de Cierre (UCI, 2016).

La siguiente tabla muestra las fases que propone AUP, la variación de dichas fases para la UCI y el objetivo de cada una de ellas.

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
Elaboración		

Construcción	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
Transición		
	Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Tabla 8 Descripción de las fases de la metodología AUP-UCI¹⁴.

Escenario 2

Teniendo que en cuenta que el escenario 2 es por el que se rige el proyecto ABCD v3.0, la investigación estará enfocada en el mismo, puesto que se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

1.8 Herramientas y tecnologías utilizadas

En el presente epígrafe se caracterizan las herramientas y tecnologías utilizadas en el desarrollo del componente para el módulo Adquisición del sistema ABCD versión 3.0.

Herramientas

1.8.1 Visual Paradigm versión 8.0

Visual Paradigm es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software (análisis y diseño, construcción, pruebas y despliegue). Permite modelar diagramas de clases, secuencia, componentes, despliegue, generar documentación, entre otros. Posee un diseño centrado en Casos de Uso, enfocado al negocio y una capacidad de ingeniería directa e inversa al integrarse con el

¹⁴ Tomado del programa de mejora Metodología de desarrollo para la Actividad productiva de la UCI.

IDE¹⁵ de desarrollo Eclipse (también permite efectuar ingeniería inversa y directa con la BD relacional PostgreSQL, Oracle, entre otras). Esta herramienta funciona sobre múltiples plataformas, es fácil de instalar y actualizar, además, es compatible con otras ediciones. Visual Paradigm brinda soporte al modelado visual con UML y posee disponibilidad de múltiples versiones en dependencia de las necesidades o condiciones del usuario (López, 2017).

1.8.2 Eclipse versión Kepler Release 4.3

Eclipse es un entorno de desarrollo integrado que facilitó las tareas de edición, compilación y ejecución de programas durante la fase de desarrollo. Este se considera una plataforma ligera, pues puede ser personalizada mediante la incorporación de *plugins*¹⁶ para proporcionar funcionalidades en dependencia de para lo que se necesite, lo que lo hace versátil y portable. Para desarrollar el componente para el módulo Adquisición se utilizan los siguientes *plugins*: Virgo Tooling, Subclipse, Spring IDE, WTP, JBoss Tools. Eclipse también provee ayuda para codificar, ejemplo de esto es: posee un compilador en tiempo real, permite completamiento de código, generación de plantillas, formateo de código, manejo de notas en código y visualización de la documentación de código (Eclipse Foundation, 2010).

1.8.3 Gestores de Bases de Datos

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a los mismos. Permite el almacenamiento, modificación, consulta de datos pertenecientes a una base de datos organizada en uno o varios ficheros y la extracción de la información en la misma, además de proporcionar herramientas para añadir, borrar, modificar y analizar los mismos. La colección de dichos datos se denomina Base de Datos, (o en inglés, Database (DB)). El equipo de desarrollo del sistema ABCD versión 3.0 decidió utilizar para la gestión de Bases de Datos (BD) J-ISIS (para los datos bibliográficos) y PostgreSQL (para los restantes datos), a continuación se caracterizan (Irles, 2013):

PostgreSQL versión 9.3

PostgreSQL es un SGBD libre. Corre en la mayoría de los Sistemas Operativos más utilizados incluyendo, GNU/Linux, OpenSuse y Windows. Cumple el principio ACID (Atomicidad, Consistencia, Integridad,

¹⁵ *Integrated Development Environment (IDE)*. En español, Entorno de Desarrollo Integrado es un programa informático compuesto por un conjunto de herramientas de programación. Puede utilizarse para uno o varios lenguajes de programación. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Tales como NetBeans y Eclipse (Vel, 2010).

¹⁶ *(Plug-in) Programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal)* (Cal, 2015).

Durabilidad) y tiene soporte completo para llaves foráneas, unión (en inglés, *joins*), vistas, subconsultas, disparadores (*triggers*), y procedimientos almacenados (en varios lenguajes), incluye la mayoría de los tipos de datos de los estándares SQL92 y SQL99, así como herencia entre tablas, por lo que se le considera un gestor de BD relacionales. Posee una documentación completa. Mediante un sistema denominado Acceso Concurrente Multiversión (por sus siglas en inglés, *Multiversion Concurrency Control* o MVCC) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*¹⁷ (Laffita, 2015).

J-ISIS versión septiembre 2014

ISIS como software, una de las funciones que desempeña es fungir como servidor para proveer acceso a la Base de Datos J-ISIS, archivo en que la información está contenida en registros numerados secuencialmente (en inglés, MFNs o *Master File Numbers*¹⁸) con valores sobre todo textuales entre otros (*integer, date, time, etc.*) almacenados en campos con una "etiqueta" (o identificador numérico) y subcampos (con identificador de un carácter). Subcampos, campos y registros son de longitud variable (se adaptan al tamaño de la información contenida en ellos) y de "ocurrencia variable", entre 0 (no presente) y cualquier número mayor de ocurrencias, en J-ISIS, sin límite. Puede manejar registros bibliográficos, junto con datos de usuarios y datos de transacción (por ejemplo, préstamos) en una única BD (Ramirez, 2012). En el módulo Adquisición del sistema ABCD versión 3.0 se utilizan dos bases de datos que permiten operar las adquisiciones, Adquisición: donde se registran los datos relacionados con las adquisiciones de materiales bibliográficos realizadas y Ejemplar: donde se registran los datos relacionados con cada ejemplar adquirido). Para la integración del componente para el módulo Adquisición con las BD J-Isis se utiliza el componente *JIsisDataProvider*.

Tecnologías

1.8.4 Eclipse Virgo versión 3.6.3

Virgo, es un servidor de aplicaciones completamente modular, basado en Java. Diseñado para ejecutar aplicaciones Java y Spring. Ofrece una plataforma para desarrollar y desplegar aplicaciones Java. Virgo

¹⁷ Una sentencia COMMIT en SQL finaliza una transacción de base de datos dentro de un sistema gestor de base de datos relacional y pone visibles todos los cambios a otros usuarios. El formato general es emitir una sentencia BEGIN WORK, una o más sentencias SQL, y entonces la sentencia COMMIT. Una sentencia COMMIT publicará cualquiera de los savepoints (puntos de recuperación) existentes que puedan estar en uso (Laffita, 2015).

¹⁸ MFNs o Master File Numbers: en español, número en el archivo maestro (Laffita, 2015).

Kernel¹⁹ es compatible con conceptos básicos de Virgo, éste está disponible como código abierto. Virgo Kernel también puede usarse independiente como una plataforma de aplicaciones OSGi (Eclipse, 2004).

Virgo dispone de (Eclipse, 2004):

- Extensión de consola de Equinox: Permite gestionar el servidor de aplicaciones Virgo y artefactos desplegados.
- Aprovisionamiento: Suministra automáticamente las dependencias de una aplicación incluyendo bundles²⁰, planes, plan de archivos y configuraciones, desde repositorios locales y remotos.
- Spring: Puede ser fácilmente configurado para utilizar una versión diferente de Spring (en el caso de ABCD versión 3.0, soporta versiones menores de Spring 4.0).

El servidor de aplicaciones Eclipse Virgo es una implementación de Eclipse. Permite la adición de bundles (posee mecanismos para gestionar la comunicación entre estos) y la configuración de planes para su ejecución.

1.8.5 RAP versión 3.0M5

Plataforma de Aplicaciones Remotas (por sus siglas en inglés, RAP), es un framework para aplicaciones modulares y Aplicaciones de Internet Enriquecidas (en inglés, Rich Internet Application (RIA²¹), desarrollado por Eclipse. RAP se integra con tecnologías de Eclipse (por ejemplo: Equinox y Virgo). Ofrece un amplio conjunto de componentes que posibilitan construir interfaces gráficas y aplicaciones (widgets²²). Estas pueden ser desarrolladas para el escritorio, el navegador de Internet, y clientes móviles. Mediante la integración de RAP con tecnologías Java los widgets pueden ser desplegados directamente como paquetes OSGi en Virgo y otros servidores de aplicaciones. Los mismos pueden ser instalados de forma remota, iniciados, detenidos, actualizados y desinstalados sin necesidad de reiniciar (Eclipse, 2015b). RAP implementa un conjunto de herramientas widget estándar (en inglés, The Standard Widget Toolkit (SWT²³) que forma parte de Eclipse, el programador al hacer uso de este, utiliza el lenguaje de programación Java,

¹⁹ Un núcleo o Kernel es un software que constituye una parte fundamental del sistema operativo, se define como la parte que se ejecuta en modo privilegiado (conocido también como modo núcleo). Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema.

²⁰ Paquetes, módulos.

²¹ Una Rich Internet Application (RIA) es una aplicación web diseñada para ofrecer las mismas características y funciones que normalmente se asocian con aplicaciones de escritorio (Busch y Koch, 2010).

²² Un widget es un elemento de una interfaz gráfica de usuario (GUI) que muestra información o proporciona una forma específica para que un usuario interactúe con el sistema operativo o una aplicación. Incluyen iconos, menús desplegables, botones, cuadros de selección, indicadores de progreso, marcas de verificación de encendido y apagado, barras de desplazamiento, ventanas, bordes de ventana (que le permiten cambiar el tamaño de la ventana), botones, forma y muchos otros dispositivos para mostrar información y para invitar, aceptar y responder a las acciones del usuario (Rose, 2013).

²³ SWT es un conjunto de herramientas widget de código abierto para Java, diseñado para proporcionar un acceso eficiente y portátil a las instalaciones de la interfaz de usuario de los sistemas operativos en los que se aplique (Eclipse, 2015a).

muy similar a como si programara para una aplicación de escritorio con SWT. De esta forma se genera el .html²⁴ y el .js (archivo de texto plano que contiene scripts de JavaScript, y que permite, ser modificado con cualquier editor de textos) (Eclipse, 2015b) desde java, el cual es generado con qooxdoo²⁵.

1.8.6 OSGi Equinox versión 3.8.2

Open Services Gateway Initiative (por sus siglas en inglés, OSGi), es una especificación modular para Java que establece las formas de crear módulos y la manera en que estos interactuarán entre sí en tiempo de ejecución. Posee diferentes implementaciones (por ejemplo: Apache Félix, Eclipse Equinox, Knoplerfish). Su objetivo es definir las especificaciones que permitan diseñar plataformas compatibles que puedan proporcionar múltiples servicios. OSGi intenta solventar los problemas del tradicional "*classloader*"²⁶ de la máquina virtual y de los servidores de aplicaciones Java. Como plataforma modular y ligera proporciona a los desarrolladores un entorno orientado a servicios y basado en componentes (Montero, 2016).

Equinox proporciona un marco de trabajo java de uso general, seguro y administrado que soporta el despliegue dinámico de aplicaciones conocidas como "*Bundles*" o módulos (Montero, 2016).

Características que componen este marco de trabajo (Montero, 2016):

- Es un sistema de módulos para la plataforma java.
- Incluye reglas de visibilidad, gestión de dependencias y versionado de los bundles
- La instalación, arranque, parada, actualización y desinstalación de bundles se realiza dinámicamente en tiempo de ejecución sin tener que detener por completo la plataforma.
- Se trata de una arquitectura orientada a servicios dentro de la Máquina Virtual.

Se pueden destacar algunos de los principales beneficios que proporciona esta tecnología:

- Reutilización del código "*out of the box*"²⁷.
- Simplificar los proyectos en los que participan muchos desarrolladores en diferentes equipos.
- Se puede utilizar en sistemas más pequeños.

²⁴ *HyperText Markup Language (HTML)* se utiliza para crear y representar visualmente una página web (Mozilla Developer Network, 2005).

²⁵ *Qooxdoo* es un marco de JavaScript universal que te permite crear aplicaciones para una amplia gama de plataformas. Con su modelo de programación orientado a objetos puede crear aplicaciones de internet enriquecidas (RIA), aplicaciones nativas para dispositivos móviles, aplicaciones web tradicionales o incluso aplicaciones para ejecutar fuera del navegador (Samuel, 2018).

²⁶ *El Java Classloader* (en español, cargador de clases Java) es una parte del Java Runtime Environment que carga dinámicamente clases Java en la Máquina Virtual de Java. El cargador de clases es responsable de localizar bibliotecas, leer sus contenidos, y cargar las clases contenidas dentro de las mismas. Esta carga es normalmente hecha "bajo demanda", por lo que no ocurre hasta que la clase sea usada por el programa. Una clase con un nombre dado sólo puede ser cargada una vez por un classloader dado (Laffita, 2015).

²⁷ La expresión en inglés "*out of the box*", es una frase que los estadounidenses utilizan con frecuencia. Significa que el código está listo para usarse. Que no necesita cambios. Que tal y como está ya es utilizable (Laffita, 2015).

- Gestiona los despliegues locales o remotos.
- Es una tecnología abierta²⁸.

Equinox es una implementación de OSGi utilizada por la plataforma de Eclipse. El modelo de programación de OSGi (Equinox) permite definir componentes de software dinámicos, es decir, los servicios OSGi, que también pueden ser parte de una aplicación basada en Eclipse (Caffarel, 2017). OSGi (Equinox) determina una arquitectura común para proveedores de servicios y desarrolladores, también permite desarrollar, desplegar y trabajar con servicios de manera coordinada.

1.8.7 Spring Dynamic Modules versión 3.1.0

Spring Dynamic Modules (por sus siglas en inglés, Spring DM) es un proyecto de Spring que permite utilizar el Framework del mismo nombre para el desarrollo de aplicaciones que pueden ser ejecutadas en la plataforma OSGi, además, se acopla a cualquier aplicación sin necesidad de modificar tu código para utilizar las funcionalidades y beneficios que ofrece y simplifica el desarrollo de paquetes OSGi utilizando conceptos y características de este, tales como la inyección de dependencias (la cual permite construir el software con poco acoplamiento) y de archivos XML²⁹ de configuración (beans-context y osgi-module-context). Entre otras funcionalidades de Spring DM se encuentran (UDLAP, 2013):

- Busca automáticamente paquetes basados en Spring y crea instancias del contexto de la aplicación Spring para cada bundle buscado.
- Busca e importa dinámicamente una referencia de servicios OSGi como beans a través de configuraciones XML.

Ventajas de la Inyección de dependencias:

- Proporciona bajo acoplamiento³⁰ entre componentes (provee componentes más desacoplados e independientes).

²⁸ El concepto de tecnología abierta hace referencia a algo que puede ser utilizado libremente sin tener que conseguir licencia o autorización para su uso (Galicia, 2015).

²⁹ XML, siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas utilizado para almacenar datos en forma legible. Permite definir la gramática de lenguajes específicos. XML da soporte a Bases de Datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información (w3schools, 2015).

³⁰ Acoplamiento es una medida de la fuerza con que un element está conectado a, tiene conocimiento de, confía en, otros elementos. Un elemento fuertemente acoplado se resiente de los cambios en los elementos relacionados, son difíciles de entender de manera aislada y difíciles de reutilizar (requiere las clases 'acopladas'). La solución es asignar responsabilidades de manera que el acoplamiento permanezca bajo (Departamento de Lenguajes y Sistemas Informáticos, 2018).

- La responsabilidad de cada uno de los componentes será muy clara y un cambio podrá ser más fácil de implementar.

1.8.8 EclipseLink versión 2.3

EclipseLink es un proyecto de código abierto que permite a los desarrolladores de Java interactuar con base de datos, Web Services, objetos XML, entre otros. La intención del proyecto es ofrecer unos estándares de solución centrada en JPA, pero con la capacidad de utilizar funciones avanzadas para aquellas aplicaciones donde se requieren. Al centrarse primero en JPA, el proyecto EclipseLink permite la integración amplia y minimiza el acoplamiento, así como proporciona un marco de persistencia que se ejecuta en cualquier ambiente Java. EclipseLink JPA ofrece soporte avanzado para contenedores de Java y bases de datos relacionales (EclipseLink, 2017).

1.9 Conclusiones del capítulo

El análisis de las características estructurales y funcionales de los protocolos de búsqueda e intercambio de información bibliográfica, permitió adoptar el estándar de interoperabilidad OAI-PMH, teniendo en cuenta el cumplimiento de las 4 libertades de software libre, lo que facilitará las modificaciones necesarias para adaptarlas a cualquier escenario de producción. Otro aspecto de relevancia fue su implementación a escala mundial.

Con el estudio de las funcionalidades del SGBD J-ISIS se indagó en el acceso a los registros bibliográficos que almacena. La descripción de los formatos de metadatos, permitió establecer equivalencia entre las etiquetas de Dublin Core y los campos del módulo Adquisición, lo que le permitirá al usuario entender la información que consultó. Como complemento de la investigación se estudiaron las herramientas y tecnologías de desarrollo de software que cuentan con los requerimientos necesarios y actualizados para el desarrollo de la aplicación.

Capítulo 2: Análisis y diseño de la propuesta de solución para el componente del módulo Adquisición del Sistema ABCD v3.0

2.1 Introducción

En el presente capítulo se describe la solución propuesta para el componente a desarrollar. Se realiza el diseño del mismo a través de los artefactos ingenieriles correspondientes a la metodología descrita anteriormente. Estos artefactos incluyen la elaboración del modelo de dominio, el planteamiento de los requisitos funcionales y no funcionales que requiere dicho componente, la elaboración y descripción de los casos de uso del sistema, las matrices de trazabilidad que requiere la metodología, así como la descripción de la arquitectura, el estilo arquitectónico y los patrones de diseño empelados.

2.2 Descripción de la propuesta de la solución

Actualmente, el Sistema REPXOS v3.0 gestiona documentos en formato digital. El bibliotecario almacena estos documentos de forma manual en el módulo Adquisición del Sistema ABCD v3.0, los cuales pasan a catalogación como documentos sin catalogar. Esto significa que el bibliotecario almacena toda la documentación registro por registro ingresando los datos en el sistema, lo que trae como consecuencia duplicidad de la información, puesto que se realiza el mismo proceso doblemente, y que en ocasiones los metadatos no estén correctos o falte alguno de ellos. Por tal motivo se implementará en el módulo Adquisición del Sistema ABCD v3.0 un componente que permita recolectar los metadatos contenidos en REPXOS v3.0 útiles para ABCD v3.0, haciendo uso de las buenas prácticas que ofrece el protocolo OAI-PMH.

Este componente permitirá que el Sistema ABCD v3.0 recolecte metadatos desde REPXOS v3.0 a través del protocolo OAI-PMH. Luego podrá mapear los metadatos recolectados que estén en el formato Dublin Core en correspondencia con los metadatos del módulo Adquisición. Guardará estos metadatos en la base de datos de Adquisición y continuará con el proceso que realiza el sistema habitualmente.

Funcionamiento del protocolo OAI-PMH

El protocolo *OAI-PMH* basa su funcionamiento en formular peticiones (denominadas en ocasiones “verbos”) y obtener respuestas, entre el Proveedor de Datos y el Proveedor de Servicios.

Está basado en la arquitectura Cliente-Servidor y sigue el principio de que existen múltiples proveedores de datos (PD) que comparten información con múltiples proveedores de servicios (PS) a través de un protocolo común. Los primeros son los depósitos de documentos que proporcionan los metadatos de los documentos

que almacenan y los segundos son los recolectores o agregadores de contenidos, que toman los datos con el objetivo de incorporarles algún valor añadido y presentarlos a los usuarios finales. Entre los valores añadidos que se pueden ofrecer existen: sistema de búsqueda e identificación, filtrado, alertas temáticas, medición del uso e impacto de los documentos, entre otros (Martín Gavilán, 2009).

OAI-PMH usa transacciones HTTP (Hypertext Transfer Protocol, por sus siglas en inglés o Protocolo de Transferencia de Hipertexto, por sus siglas en español), que es el protocolo usado en cada transacción de la World Wide Web desde un cliente o recolector hacia un servidor o archivo en forma de preguntas.

Ejemplo: el cliente emite una solicitud al servidor pidiéndole le envíe metadatos con los datos que contenga, como respuesta el segundo devuelve un XML donde se incluyen los datos de la petición (Velandia, 2007).

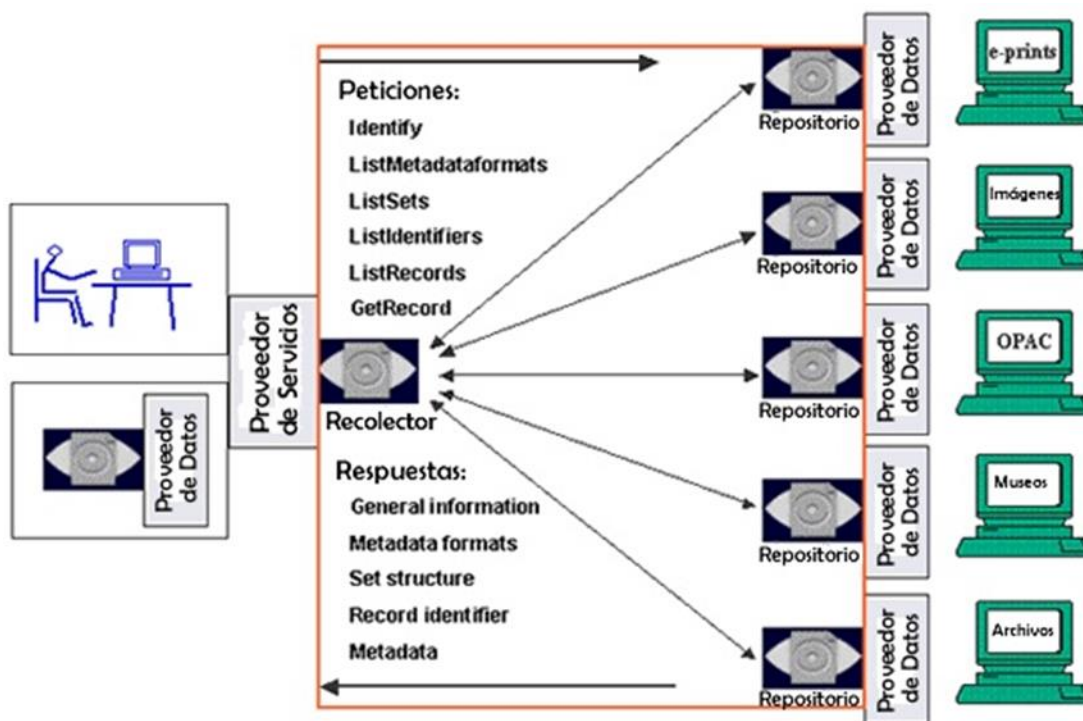


Ilustración 4 Funcionamiento del protocolo OAI-PMH³¹.

El usuario envía peticiones OAI-PMH a través del método **GET** o un Uniform Resource Locator (URL, por sus siglas en inglés) sobre el navegador web a un proveedor de servicios (generalmente son servicios web), el cual solicita a un proveedor de datos que le envíe registros de metadatos de diferentes recursos con que este disponga. Para ello el proveedor de datos extrae la información bibliográfica necesaria según las

³¹ <https://www.oaforum.org/tutorial/english/page3.htm>.

especificaciones establecidas en la petición OAI-PMH y luego envía la respuesta al proveedor de servicios, como un conjunto de registro de metadatos en formato Extensible Markup Language (XML, por sus siglas en inglés o Lenguaje de Marcas Extensible, por sus siglas en español), el cual mostrará los resultados al usuario, por medio de una interfaz (Velandia, 2007).

Existen seis verbos o procedimientos definidos en OAI-PMH, los cuales se pueden invocar a un proveedor de datos (Dueñas, 2005):

Lista de verbos OAI-PMH	
Verbo	Función
Metadatos acerca del repositorio	
Identify	Descripción del repositorio
ListMetadataFormats	Lista de todos los tipos de metadatos soportados en el repositorio
ListSets	Lista de las particiones lógicas de información en el repositorio
Verbos de recolección de metadatos	
ListIdentifiers	Lista de los identificadores OAI únicos existentes en el repositorio
ListRecords	Lista de n registros de metadatos
GetRecord	Lista de un solo registro de metadatos

Tabla 9 Lista de verbos OAI-PMH³².

³² Para mayor información esta lista de verbos se encuentra disponible en la dirección <http://eprints.rclis.org/9539/1/OAI.pdf>.

2.3 Modelo de dominio

“El modelo de dominio captura los tipos de objetos de mayor relevancia en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema” (Sommerville, 2011). Este es el principal motivo por el que el modelo de dominio puede ser tomado como punto inicial para el diseño del sistema. Puede utilizarse para capturar y expresar los conceptos más importantes del contexto del mismo. Es una representación visual de los conceptos u objetos del mundo real, significativos para un problema o área de interés. Los analistas utilizan el modelo de dominio como un medio para comprender los procesos de negocios donde el sistema va a ser utilizado.

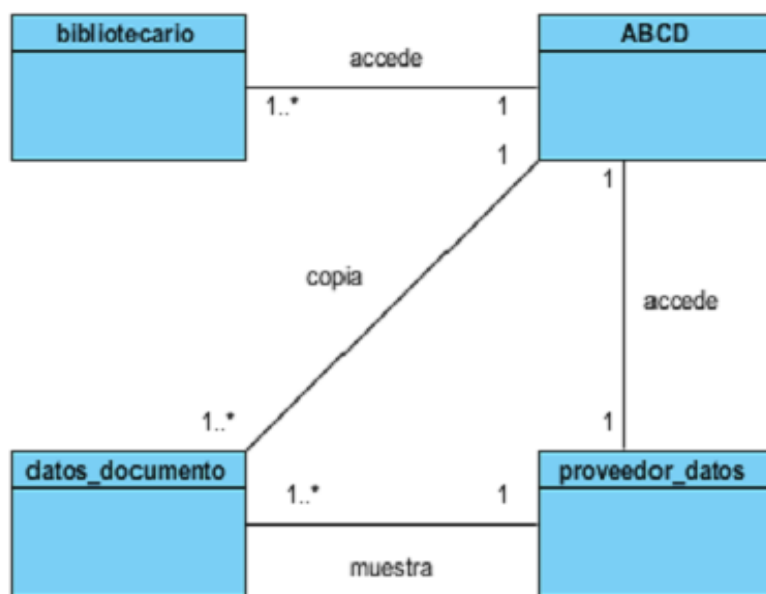


Ilustración 5 Diagrama de clases del modelo de dominio.

Fuente: Elaboración propia

Actualmente, el proceso de recolección ocurre de la siguiente forma: el bibliotecario accede por un navegador web al repositorio institucional donde busca el documento que desea recolectar, luego copia los datos del recurso deseado con los cuales recolecta el documento en el SIGB.

2.4 Especificación de los requisitos del software

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 (Sommerville, 2011).

2.4.1 Requerimientos funcionales

Los requisitos funcionales son capacidades o condiciones con las cuales el sistema debe cumplir e indican su comportamiento, es decir son las cosas que el sistema puede hacer. Cuando se expresan como requerimientos del usuario, habitualmente se describen de forma general mientras que los requerimientos funcionales del sistema describen con detalle la función de este, sus entradas y salidas, excepciones. A continuación, se listan y describen los requisitos funcionales que debe cumplir el sistema para la solución planteada.

- RF1: Adicionar proveedor.
Descripción: Permite al administrador agregar un proveedor de datos.
- RF2: Eliminar proveedor.
Descripción: Permite al administrador eliminar un proveedor de datos.
- RF3: Modificar proveedor.
Descripción: Permite al administrador modificar un proveedor de datos.
- RF4: Listar proveedores.
Descripción: Permite al administrador ver los proveedores de datos.
- RF5: Recolectar metadatos por colección.
Descripción: Permite al administrador recolectar los datos de una colección dada, permite especificar rangos de fecha.
- RF6: Mostrar las colecciones de un repositorio.
Descripción: Permite al bibliotecario ver las colecciones que posee un repositorio determinado.
- RF7: Comprobar proveedor de datos.
Descripción: Permite al bibliotecario comprobar los datos del repositorio que seleccione.

2.4.2 Técnicas de validación de requisitos

El objetivo fundamental de la validación de requisitos es demostrar que la definición de los mismos es lo que realmente el usuario necesita o el cliente desea. Luego de realizar la captura de requisitos se procede a la validación de estos. Los errores de requisitos en el documento pueden conducir a grandes costos, como repetir el trabajo cuando son descubiertos durante el desarrollo o después de que el sistema esté en uso, por lo que es de suma importancia validarlos.

Para validar los requisitos de este trabajo, se utilizó la técnica de generación de casos de prueba. En ella los requerimientos deben poder probarse. Si las pruebas para estos se conciben como parte del proceso de validación, a menudo revela problemas en los requerimientos. Para esta investigación, fueron realizados diseños de casos de pruebas a todos los requerimientos del sistema, los cuales arrojan diferentes tipos de errores; como de validación, ortográficos y de interfaz.

2.4.3 Requerimientos 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 este, como la fiabilidad y el tiempo de respuesta” (Sommerville, 2011). En resumen, los requisitos no funcionales representan aquellas cualidades que debe poseer el sistema pero que no son funcionalidades específicas.

Los requerimientos no funcionales que debe tener el componente a desarrollar son:

Requerimientos no funcionales de software

Usabilidad:

- RNF1: El sistema podrá ser utilizado con el fin de cosechar metadatos desde cualquier proveedor de datos que cumpla con las especificaciones que establece el protocolo OAI-PMH.

Soporte

- RNF2: Se debe implementar el sistema siguiendo el estándar de codificación.
- RNF3: Los componentes de software que integran la solución se organizarán de forma modular.

Seguridad

- RNF4: Para acceder al módulo los usuarios deben de estar autenticados.

Portabilidad

- RNF5: El sistema debe ser capaz de ejecutarse sobre plataforma Linux.

Legales

- RNF6: Las herramientas seleccionadas para el desarrollo del módulo están respaldadas por licencias libres o GPL.

Requerimientos no funcionales de hardware (servidor)

- RNF7: Requisitos del sistema operativo: Linux.
- RNF8: Máquina Virtual de Java (JVM) 1.7 (implementación de Oracle) y PostgreSQL 9.3 o superior.

Requerimientos no funcionales de hardware (equipos clientes)

- RNF9: Navegador: Mozilla Firefox 38+ o Google Chrome.
- RNF10: Requisitos de pantalla: resolución mínima de 1360 x 768.

2.4.4 Listado de casos de uso del sistema

- CU_1 Gestionar proveedores.
- CU_2 Recolectar metadatos por colección.
- CU_3 Mostrar colecciones de un repositorio.
- CU_4 Comprobar proveedor de datos.

2.5 Modelo de casos de uso del sistema

El diagrama de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones. Describe lo que hace el sistema para cada tipo de usuario y permite que los desarrolladores del mismo y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir dicho sistema (Sommerville, 2011).

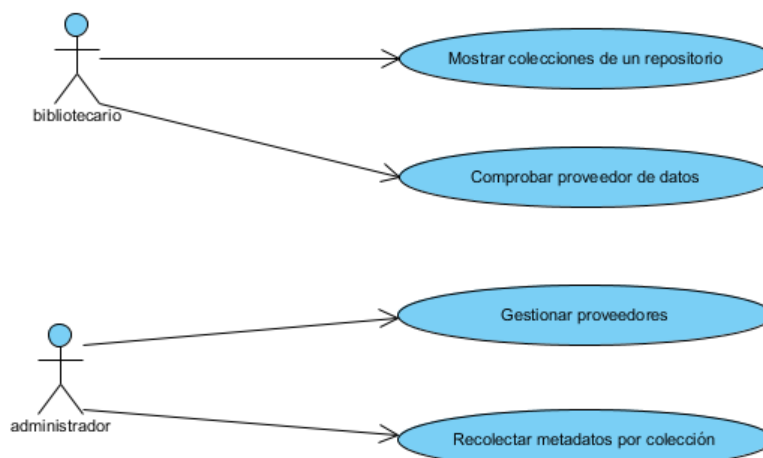


Ilustración 6 Diagrama de casos de uso del sistema.

Fuente: Elaboración propia

2.5.2 Patrón de casos de uso

Para reflejar requisitos reales con mayor precisión, facilitar el trabajo con los sistemas y simplificar su mantenimiento se optó por la utilización de patrones de casos de usos. La comunidad de desarrollo de software adoptó estas técnicas para que a través de herramientas se lograra resolver de forma ágil y sistemática los problemas que se les plantean a los desarrolladores.

Estos patrones están enfocados en el diseño y las técnicas utilizadas en modelos de alta calidad y no en cómo modelar usos específicos. Con el empleo de los mismos, arquitectos, analistas, ingenieros, y gerentes pueden lograr resultados eficientes de manera más rápida.

CRUD es uno de estos patrones, el cual está presente en los casos donde se quiere realizar inserciones, eliminaciones, modificaciones y consultas a alguna entidad del sistema. Su nombre es un acrónimo de las palabras en inglés Create, Read, Update, Delete. El patrón CRUD Completo consiste en un caso de uso para administrar la información de una entidad que permite modelar diferentes operaciones tales como crear, leer, cambiar y eliminar o dar de baja (Alvarez, 2013).

A continuación, se muestra el caso de uso gestionar repositorio OAI-PMH donde se evidencia la utilización de dicho patrón en la propuesta de solución.

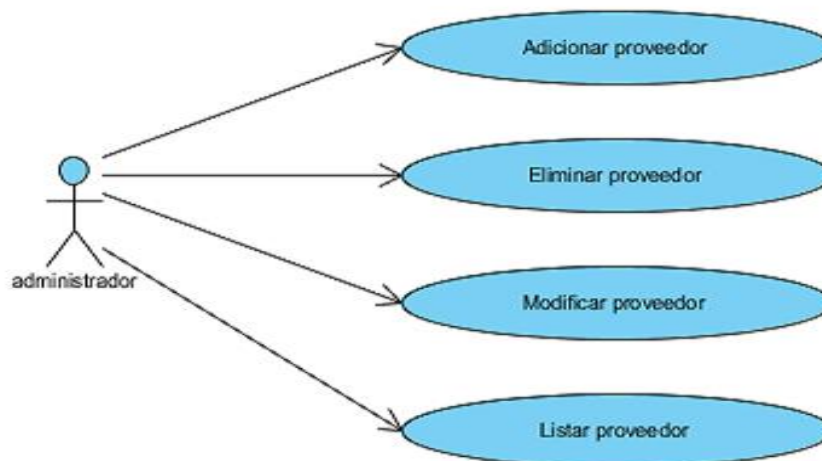


Ilustración 7 Diagrama de caso de uso Gestionar proveedores antes de aplicar el CRUD completo.

Fuente: Elaboración propia

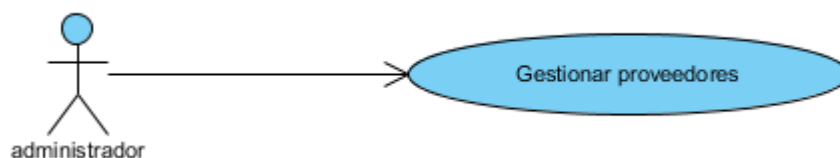


Ilustración 8 Diagrama de casos de uso Gestionar proveedores después de aplicar el CRUD completo.

Fuente: Elaboración propia

2.5.3 Actores del sistema

Los actores del sistema representan entidades externas que interactúan directamente con el sistema (personas, máquinas u otros sistemas) (Sommerville, 2011). A continuación, se muestran los actores que interactúan con el sistema:

Actor	Descripción
Administrador	Es la persona que gestiona los proveedores de datos y realiza las peticiones OAI.
Bibliotecario	Se encarga de recolectar los registros.

Tabla 10 Actores del sistema.

Fuente: Elaboración propia

2.5.4 Matriz de trazabilidad

La matriz de trazabilidad relaciona dos elementos esenciales para la buena ejecución de las labores de un proyecto: los requisitos establecidos para dicha ejecución y el valor que cada uno de ellos agrega al conjunto del proceso. Es una herramienta clave para la ingeniería de los proyectos, así como para el seguimiento de los diversos elementos que los componen (OBS, 2018).

A continuación, se muestra la matriz de trazabilidad de los requerimientos con los casos de uso (Req_CU), la cual establece la relación entre ellos. Las restantes matrices se encuentran anexadas al final del documento. [\(Ver anexo 2\)](#)

(4) Use Case		Comprobar proveedor de ...	Gestionar proveedores	Mostrar colecciones de un ...	Recolectar metadatos por ...
By: Transitor		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
(7) Requirement					
Adicionar proveedor			<input checked="" type="checkbox"/>		
Comprobar proveedor de ...		<input checked="" type="checkbox"/>			
Eliminar proveedor			<input checked="" type="checkbox"/>		
Listar proveedor			<input checked="" type="checkbox"/>		
Modificar proveedor			<input checked="" type="checkbox"/>		
Mostrar colecciones de un ...				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Recolectar metadatos por ...					<input checked="" type="checkbox"/>

Ilustración 9 Matriz de trazabilidad Req_CU.

Fuente: Elaboración propia

2.5.5 Descripción de los casos de uso

Para una mejor comprensión de la funcionalidad asociada a cada caso de uso es necesario describir cada uno de estos. Dicha descripción puede ser elaborada de forma breve o extendida y debe ir acompañada del prototipo de interfaz de usuario respectivo. Los casos de uso son artefactos narrativos que describen bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario.

A continuación, se realiza la descripción del caso de uso Gestionar proveedores de prioridad media y la descripción del caso de uso de Recolectar metadatos por colección de prioridad crítica.

Caso de uso	Gestionar proveedores.
Actores	Administrador.

Resumen	El caso de uso se inicia cuando el administrador selecciona la opción Proveedor de URL y elige adicionar, eliminar, modificar, o listar un proveedor de datos. El sistema solicita los datos en dependencia de la opción seleccionada, el administrador inserta un nuevo proveedor, realiza las modificaciones deseadas, elimina el proveedor y puede ver los proveedores existentes. El sistema valida y guarda los datos, finalizando así el caso de uso.
Precondiciones	El administrador debe estar autenticado.
Referencias	RF1, RF2, RF3, RF4.
Prioridad	Media.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Proveedor de URL.	2. El sistema muestra los datos de los proveedores de datos almacenados en una tabla y las opciones: adicionar proveedor, editar y eliminar. a) Si selecciona la opción adicionar proveedor (ir a la sección "Adicionar proveedor OAI-PMH"). b) Si selecciona la opción editar (ir a la sección "Editar"). c) Si selecciona la opción eliminar (ir a la sección

	<p>“Eliminar”).</p> <p>d) Si selecciona la opción comprobar (ir a la sección “Comprobar”).</p>
--	--

Flujo Normal de Eventos

Sección Adicionar proveedor OAI-PMH

Acción del actor	Respuesta del sistema
1. Selecciona la opción Proveedor de URL.	
	2. Muestra una interfaz donde pide los datos.
3. Introduce los datos del nuevo proveedor.	
4. Selecciona la opción Guardar.	
	5. Valida los datos.
	6. Informa que se guardó el nuevo proveedor.

Flujo alterno Campos incorrectos

Acción del actor	Respuesta del sistema
	5.1 Muestra un mensaje indicando los campos errados para su corrección (Ir a la acción 2).

Sección Editar

Acción del actor	Respuesta del sistema
------------------	-----------------------

1. Selecciona la opción Editar.	2 Muestra una interfaz donde el administrador puede cambiar los datos.
3. Edita los datos del proveedor.	
4 Selecciona la opción Guardar.	
	5. Valida los datos.
	6. Informa que se guardó el nuevo proveedor.
Flujo alternativo Campos incorrectos	
Acción del actor	Respuesta del sistema
	5.1 Muestra un mensaje indicando los campos errados para su corrección (Ir a la acción 2).
Sección Eliminar	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Eliminar.	2. Muestra una interfaz donde con el nombre del servidor y pide que confirme si desea eliminarlo.
3. Selecciona Si deseo eliminar el servidor.	
	4. Elimina el servidor.
	5. Muestra una interfaz informando que se eliminó el servidor.
Sección Consultar	

Acción del actor	Respuesta del sistema
1. Selecciona la opción Consultar.	
	2. Muestra los datos de los proveedores guardados en el sistema.

Tabla 11 Descripción del caso de uso Gestionar proveedores.

Fuente: Elaboración propia

Caso de uso	Recolectar metadatos por colección.
Actores	Administrador.
Resumen	El caso de uso se inicia cuando el administrador decide recolectar los datos de una colección específica de un repositorio.
Precondiciones	El administrador debe estar autenticado.
Referencias	RF5.
Prioridad	Crítica.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema

1. Selecciona el proveedor de datos.	
	2. El sistema lista las colecciones correspondientes al servidor seleccionado.
3. Selecciona la colección que desea recolectar.	
4. Selecciona los rangos de fecha (desde, hasta).	
5. Selecciona la opción Recolectar.	
	6. Valida los datos.
	7. El sistema realiza la petición al servidor.
	8. El sistema informa el éxito de la recolección.
Flujo alternativo Error de fecha	
Acción del actor	Respuesta del sistema
3.1 Selecciona la fecha dese (mayor) a la fecha hasta.	5.1 Informa que la fecha de desde debe ser anterior a la fecha hasta.
3.2 Selecciona la fecha hasta (menor) a la fecha desde.	5.2 Informa que la fecha de hasta debe ser mayor a la fecha desde.
Flujo alternativo	
Acción del actor	Respuesta del sistema

	8.1 El sistema informa que no se pudo realizar la recolección (Ir a la acción 2).
--	---

Tabla 12 Descripción del caso de uso Recolectar metadatos por colección.

Fuente: Elaboración propia

2.6 Descripción de la arquitectura

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos. Es una representación que permite analizar la efectividad del diseño para cumplir los requerimientos establecidos, considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y reducir los riesgos asociados con la construcción del software (Pressman, 2010).

2.6.2 Estilo arquitectónico

Los estilos arquitectónicos son transformaciones impuestas al diseño de todo un sistema. El objetivo es establecer una estructura para todos los componentes del sistema (Pressman, 2010).

Aunque se han creado millones de sistemas de cómputo en los últimos 50 años, la gran mayoría puede clasificarse en un número relativamente pequeño de estilos arquitectónicos: Arquitectura Centrada en Datos, Arquitectura de Flujos de Datos, Arquitectura de Llamada y Retorno, Arquitectura Orientada a Objetos y Arquitectura Estratificada (Pressman, 2010).

En la presente investigación se utiliza el estilo arquitectónico Llamada y Retorno. El mismo permite que un diseñador de software (arquitecto del sistema) obtenga una estructura de programa que resulte relativamente fácil modificar cuando se considere necesario. Este estilo se aplica en sistemas de gran escala que estén encaminados a reflejar la estructura del lenguaje de programación, basándose en la abstracción de procedimientos, funciones o métodos.

Específicamente se hace uso del estilo Arquitecturas en Capas, el cual descompone la aplicación en cuantas capas sean necesarias para distribuir las subtarear, donde cada capa proporciona servicios a la capa inmediata superior y se sirve de las prestaciones que le brinda la inmediata inferior. La arquitectura definida para el desarrollo de la solución se encuentra dividida en cuatro capas, que a continuación se explican (Brown y Arias, 2015).

La siguiente ilustración muestra la arquitectura OSGI del sistema ABCD v3.0.

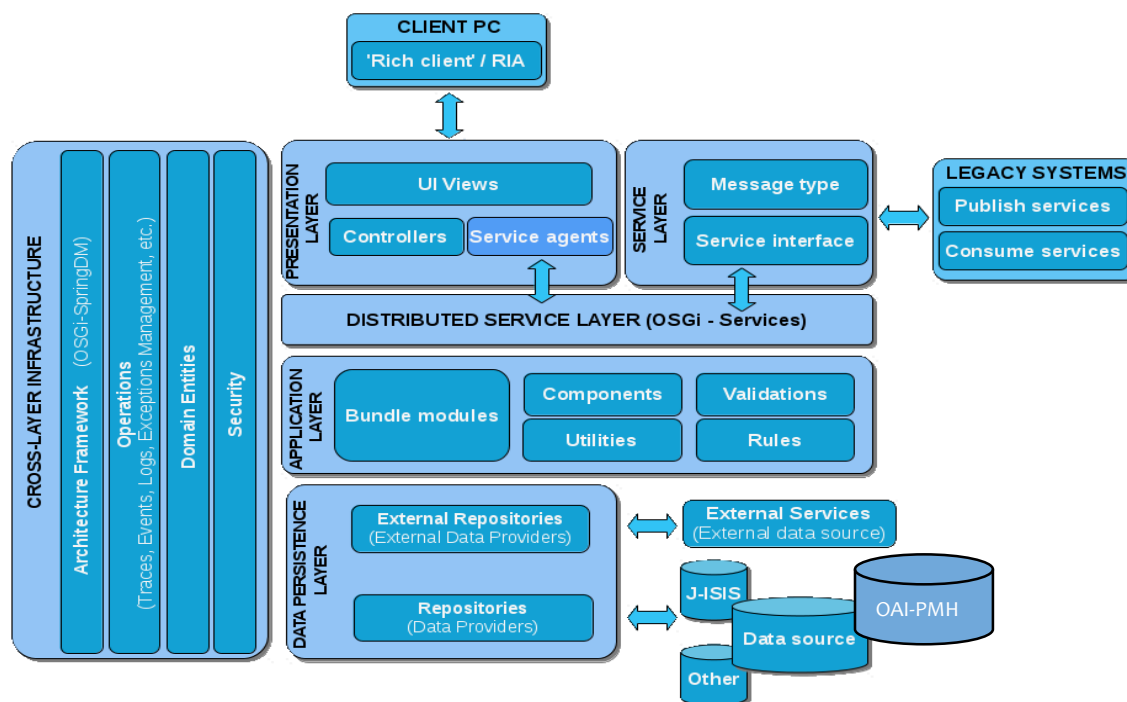


Ilustración 10 Arquitectura OSGI³³.

1. Capa de Presentación:
 - Está compuesta por las interfaces de usuario, el controlador y el agente de servicio.
2. Capa de Lógica de negocio:
 - Componentes que modelan la lógica de negocio.
 - Reciben acciones de la capa de presentación y utilizan la capa de datos para manipular información. Integración sencilla y eficaz con sistemas externos.
3. Capa de Acceso a datos:
 - Almacenamiento, actualización y consulta de los datos del sistema.
 - Se agrega soporte para una nueva base de datos en período corto.
 - Puede estar en el mismo servidor de la lógica de negocio o distribuida
4. Capa de Infraestructura:
 - Añade gestión dinámica de permisos proporcionando un modelo de seguridad capaz de autenticar el código a través de la ubicación del *bundle* y de sus paquetes.

³³ Tomado del trabajo de diploma: *Diseño e implementación del módulo Catálogo en Línea de Acceso Público del sistema ABCD 3.0* (Brown y Arias, 2015).

- Provee una infraestructura para desplegar y manejar aplicaciones que tienen que ejecutarse en un entorno seguro y controlado.

2.6.3 Patrones de diseño GRASP

Un patrón de diseño es una descripción de un problema y la solución, a la que se le da un nombre y que se puede aplicar a nuevos contextos. Es decir, que describe clases y objetos comunicándose entre sí, siendo éstos adaptados para resolver un problema de diseño general. Muchos patrones proporcionan guías sobre el modo en el que deberían asignarse las responsabilidades a los objetos (Larman, 2003).

General Responsibility Assignment Software Patterns (GRASP, por sus siglas en inglés) describen los principios fundamentales del Diseño Orientado a Objetos (DOO, por sus siglas en español) y la asignación de responsabilidades expresados como patrones (Larman, 2003).

En la presente investigación se utilizan los patrones GRASP teniendo en cuenta que constituyen el fundamento de cómo se diseñará el sistema, es decir, una serie de buenas prácticas de aplicación recomendable en el diseño de software. A continuación, se ejemplifican los que serán empleados en la misma:

- **Patrón Experto:** en términos de la Programación Orientada a Objetos (POO, por sus siglas en español), es la clase que tiene la responsabilidad de conocer toda la información de las otras. Esta característica se evidencia en la clase OAI-PMH.java.
- **Patrón Creador:** en términos de POO, la clase “B” que tiene la responsabilidad de crear una instancia de la clase “A”. La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. En consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ella. El diseño, bien asignado, poder soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y la reutilización. Esta característica se evidencia en la clase OAIPMH.java. Esta clase se comunica con la clase controladora AllManagementController.java y crea nuevas instancias de las peticiones OAI-PMH en correspondencia a las solicitudes que realiza el usuario.
- **Patrón Alta Cohesión:** asignar una responsabilidad de modo que la cohesión siga siendo alta. En términos de POO, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Una clase con baja cohesión no es conveniente ya que son difíciles de comprender, de reutilizar y de conservar. Estas características

se evidencian en las clases-peticiones GetRecord.java, ListIdentifiers.java y ListRecords.java; ListIdentifiers.java pide los encabezados y ListRecords.java es listado de la clase GetRecord.java, respectivamente.

- **Patrón Bajo Acoplamiento:** tiene la responsabilidad de mantener bajo acoplamiento. En términos de POO, el acoplamiento es la medida con que un objeto (clase) está relacionado o conectado con otros. Una clase con bajo acoplamiento no depende de muchas otras. Una clase con alto acoplamiento recurre a muchas otras. Este tipo de clases no es conveniente ya que si ocurren cambios de codificación pueden afectarse una a otras. Esta característica se evidencia en las clases-peticiones Identify.java, ListSets.java y ListMetadataFormats.java.
- **Patrón Controlador:** en términos de POO, es la clase que tiene la responsabilidad de gestionar todos los eventos del sistema. Además, es el responsable directo de la interacción usuario aplicación. Esta característica se evidencia en la clase AllManagementController.java ya que se encarga de atender todas las peticiones OAI-PMH realizadas por el usuario mediante el método GET.

2.6.4 Patrones de diseño GoF

Los patrones de diseño **GoF**³⁴ se clasifican en 3 grandes categorías basadas en su propósito: creacionales³⁵, estructurales³⁶ y de comportamiento³⁷. (Angelfire, 2011). En la implementación del componente para el módulo Adquisición del Sistema ABCD versión 3.0 se utilizó el siguiente patrón perteneciente a la categoría estructural.

- **Proxy:** mantiene un representante de un objeto.

2.7 Modelo de diseño

El diseño del software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo. En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la

³⁴ Fue por los años 1994, que apareció el libro "Design Patterns: Elements of Reusable Object Oriented Software" escrito por los ahora famosos **Gang of Four** (GoF, que en español es la pandilla de los cuatro) formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Ellos recopilaron y documentaron 23 patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos. Desde luego que ellos no son los inventores ni los únicos involucrados, pero fue luego de la publicación de ese libro que empezó a difundirse con más fuerza la idea de patrones de diseño (Angelfire, 2011).

³⁵ Los patrones creacionales tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados (Angelfire, 2011).

³⁶ Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos (Angelfire, 2011).

³⁷ Los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos (Angelfire, 2011).

arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software (Pressman, 2010).

2.7.2 Diagrama de clases del diseño

Un Diagrama de Clases de Diseño muestra la especificación para las clases software de una aplicación. Incluye la siguiente información (UNAD, 2018):

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Navegabilidad.
- Dependencias.

La siguiente ilustración muestra el diagrama de clases del diseño del caso de uso Gestionar proveedor.

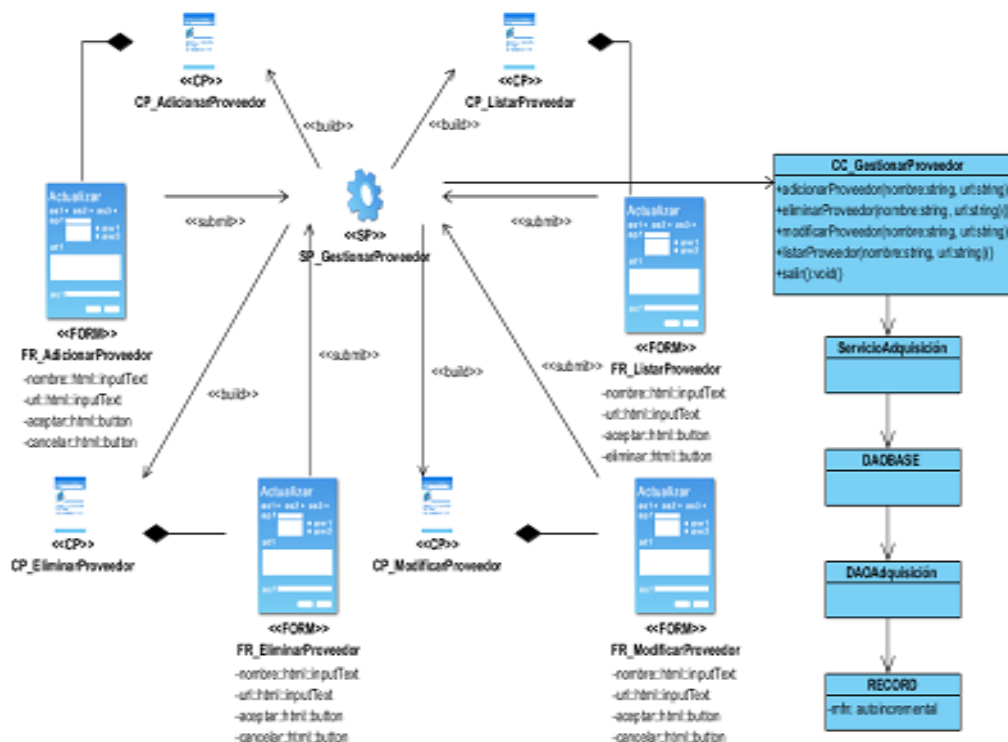


Ilustración 11 Diagrama de clases del diseño CU_Gestionar proveedores.

Fuente: Elaboración propia

Descripción de las clases

- Clase ServicioAdquisicion

Propósito

Implementar los métodos de la capa de negocio que se necesiten.

Descripción

ServicioAdquisicion
+salvar(objeto : Objeto) : void
+actualizar(objeto : Objeto) : void
+obtener(idObjeto : Objeto) : Objeto
+listarTodas(tipoObjeto : string) : list<Objeto>

Ilustración 12 Clase ServicioAdquisicion.

Fuente: Elaboración propia

- Clase DAOBASE

Propósito

Proveer los métodos comunes de acceso a dato a las capas superiores de la aplicación.

Descripción

DAOBASE
+salvar(objeto : Objeto) : void
+actualizar(objeto : Objeto) : void
+obtener(idObjeto : Objeto) : Objeto
+listarTodas(tipoObjeto : string) : list<Objeto>

Ilustración 13 Clase DAOBASE.

Fuente: Elaboración propia

- Clase DAOAdquisicion

Propósito

Proveer los métodos específicos del módulo a las capas superiores de la aplicación.

Descripción

DAOAdquisicion
+salvar(objeto : Objeto) : void
+actualizar(objeto : Objeto) : void
+obtener(idObjeto : Objeto) : Objeto
+listarTodas(tipoObjeto : string) : list<Objeto>

Ilustración 14 Clase DAOAdquisicion.

Fuente: Elaboración propia

- Clase Record

Propósito

Almacenar la información de las entidades.

Descripción



Ilustración 15 Clase Record.

Fuente: Elaboración propia

2.8 Conclusiones del capítulo

En este capítulo se especificaron las clases del dominio para lograr entender los procesos que se van a llevar a cabo en la solución, se identificó al bibliotecario y al administrador como actores, así como 7 requisitos funcionales y 10 requisitos no funcionales que se utilizarán en la elaboración del componente para el módulo Adquisición del sistema ABCD 3.0, y que a su vez fueron agrupados y representados mediante un diagrama de caso de uso con la descripción breve de los mismos. Se realizaron las matrices de trazabilidad propuestas por la metodología. Se describió la arquitectura de software definida, lo que facilitará el desarrollo del componente, así como el estilo y los patrones de diseño a emplear. Teniendo en cuenta esta arquitectura se procedió al diseño del componente donde se realizó el diagrama de clases con estereotipos web usando como base el CU Gestionar proveedores, quedando las bases sentadas para avanzar al próximo flujo de trabajo que es implementación y prueba.

Capítulo 3: Implementación y pruebas del componente del módulo Adquisición del Sistema ABCD v3.0

3.1 Introducción

En el presente capítulo se abordará el flujo de trabajo del análisis y diseño de la solución propuesta. Se hace referencia a la implementación del componente para el módulo Adquisición, basándose en el resultado del diseño realizado en el capítulo anterior. Como parte de la implementación se muestra el diagrama de despliegue. Además, se incluye el diseño de pruebas y los resultados obtenidos de las mismas en la aplicación.

3.2 Implementación

La disciplina Implementación que propone la metodología AUP-UCI, muestra cómo, a partir de los resultados del Análisis y Diseño se construye el sistema (UCI, 2016).

A partir de los resultados de diseño obtenidos fue realizada la implementación del componente a través de ficheros de código fuente, scripts, ejecutables, etc. Esta disciplina explica cómo desarrollar, organizar e integrar el componente implementado al módulo Adquisición basándose en las especificaciones del diseño.

La finalidad del componente es:

- Recolectar metadatos desde REPOS v3.0 a través del protocolo OAI-PMH.
- Mapear los metadatos recolectados que estén en el formato Dublin Core en correspondencia con los metadatos del módulo Adquisición.
- Guardar los metadatos en la base de datos de Adquisición y continuar con el proceso que realiza el sistema habitualmente.

3.3 Diagrama de despliegue

El Diagrama de Despliegue, es un tipo de diagrama del Lenguaje Unificado de Modelado (UML), que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Describe la arquitectura física del sistema durante la ejecución, en términos de (Laffita, 2015):

- Procesadores.
- Dispositivos.
- Componentes de software.

Este diagrama describe, además, la topología del sistema: la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos (Laffita, 2015).

La siguiente vista de despliegue muestra las relaciones físicas de los distintos nodos que componen el sistema y la distribución de los componentes sobre dichos nodos.

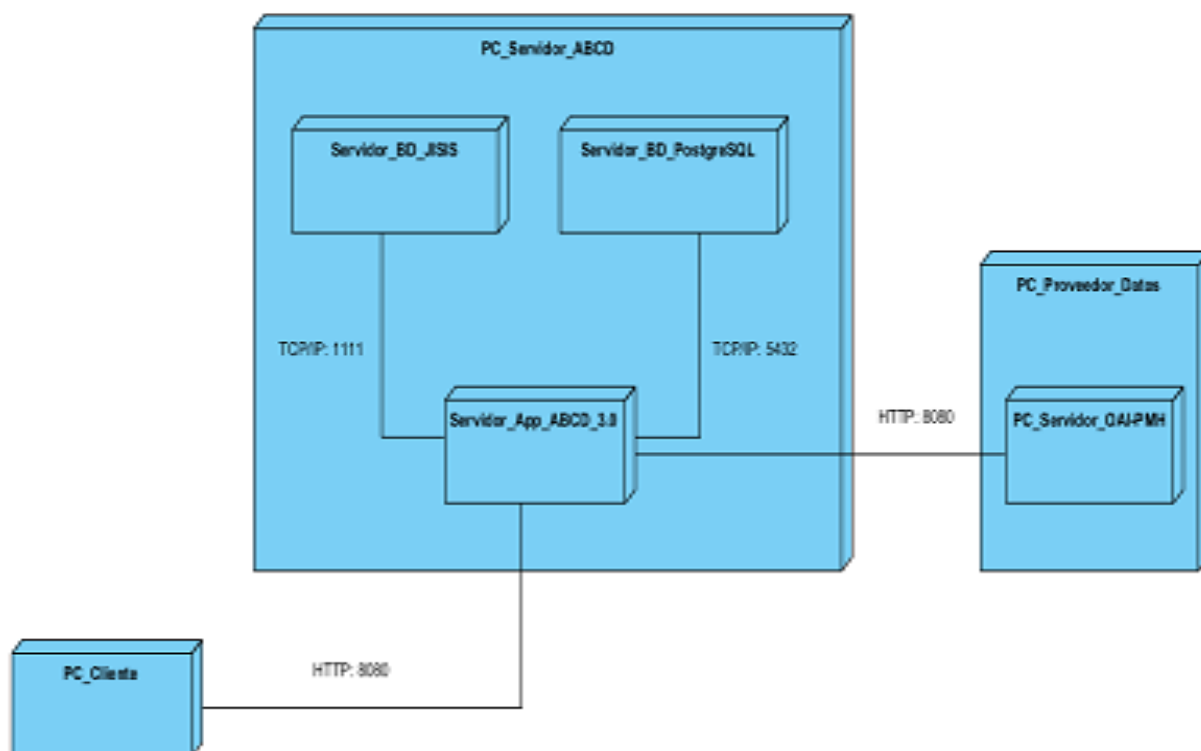


Ilustración 16 Diagrama de despliegue.

Fuente: Elaboración propia

Para el despliegue de la aplicación se necesita una PC cliente, que debe tener instalado un navegador web, el cual hace peticiones al servidor de aplicaciones Eclipse Virgo versión 3.6.3, donde se encuentra la aplicación incluyendo los *bundles* *cu.uci.abcd.acquisition.ui*, *cu.uci.abcd.acquisition* y *cu.uci.abcd.acquisition.impl*, usando el protocolo de comunicación (HTTP) por el puerto 8080. El sistema establece comunicación con los servidores de Bases de Datos PostgreSQL versión 9.3 utilizando el protocolo TCP/IP por el puerto 5432 y J-ISIS versión septiembre 2014 utilizando el protocolo TCP/IP por el puerto 1111. El servidor de aplicaciones Eclipse Virgo hace peticiones OAI-PMH al proveedor de datos a través del protocolo HTTP por el puerto 8080 que tiene incluido el servidor OAI-PMH.

3.4 Pruebas de software

Las pruebas de software (*Software Testing*, por sus siglas en inglés) comprenden el conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación, por medio de pruebas sobre el comportamiento del mismo (Lara, 2013).

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

- Estrategia de prueba.
- Niveles de Prueba.
- Tipo de prueba.
- Método de prueba.
- Caso de prueba.

3.4.1 Estrategia de pruebas

Las estrategias de pruebas de software proporcionan una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier estrategia de prueba debe incorporar la planificación, diseño, ejecución, recolección y evaluación de los casos de prueba (Pressman, 2010).

Para realizar una correcta estrategia de pruebas se deben tener en cuenta algunos criterios de importancia (Ruiz y Vaillant, 2015):

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

3.4.2 Niveles de pruebas

La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Se distinguen los siguientes niveles de pruebas (Ruiz y Vaillant, 2015):

- Prueba de desarrollador.
- Prueba independiente.

- Prueba de unidad.
- Prueba de integración.
- Prueba de regresión.
- Prueba de sistema.
- Prueba de aceptación.

Se trabajará en los siguientes niveles:

- En el nivel de sistema con el objetivo de probar las funcionalidades del componente para el módulo Adquisición, esto se hará mediante el método de caja negra empleando la técnica de partición de equivalencia.
- Aplicando las pruebas de regresión para verificar que una vez que termine una iteración y se corrijan no conformidades los nuevos cambios no afecten la calidad de lo que se había hecho antes previniendo que se cometan nuevos errores.
- En el nivel de aceptación para ver si el desarrollo del componente contribuyó o no a la recuperación de información de los usuarios ([Ver anexo 3](#)).

3.4.3 Tipos de pruebas

Existen varios tipos de pruebas que se deben tener en cuenta para un correcto de la misma (Ruiz y Vaillant, 2015):

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

El tipo de prueba a aplicar es Funcionalidad. Dentro de las pruebas de Funcionalidad se aplicarán pruebas de función. Estas pruebas fijan su atención en la validación de las funciones, métodos, servicios y casos de uso. Permiten comprobar el correcto funcionamiento de los requisitos funcionales del componente.

3.4.4 Método de pruebas

Se explica a continuación el método de prueba a emplear en la solución del componente (Ruiz y Vaillant, 2015):

Prueba de Caja Negra

- Pruebas que se llevan a cabo sobre la interfaz del software.
- El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene (no se ve el código).

Para desarrollar estas pruebas existen varias técnicas:

- Técnica de partición de equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica de análisis de valores límites: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de causa-efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

La técnica a emplear para desarrollar las pruebas es la de partición de equivalencia. Esta técnica permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico (Ruiz y Vaillant, 2015).

3.4.5 Casos de pruebas

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. Siempre es ejecutada como una unidad, desde el comienzo hasta el final (Ruiz y Vaillant, 2015).

Debe verificar:

- Si el producto satisface los requerimientos del usuario, tal y como se describe en las especificaciones de los requerimientos.
- Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

Se describieron los casos de pruebas para cada caso de uso. A continuación, se presenta el caso de prueba para el caso de uso Gestionar proveedores, especificando el escenario, la descripción, las variables, la respuesta del sistema y el flujo central, así como los resultados obtenidos una vez ejecutado el caso de prueba y las condiciones que deben cumplirse para que este se ejecute. El resto de los casos de pruebas se encuentran anexados al final del documento [\(Ver anexo 4\)](#). Las celdas de la tabla contienen los valores

V, I, NA. V indica inválido, I indica inválido y NA, indica que no es necesario proporcionar un valor del dato ya que es irrelevante.

Diseño de casos de pruebas Gestionar proveedores

Descripción general:

Permite gestionar todos los proveedores de datos.

Condiciones de ejecución:

El usuario debe estar autenticado en el sistema y poseer el rol de administrador.

SC 1 Adicionar proveedor

Escenario	Descripción	Nombre	URL	Respuesta del sistema	Flujo central
EC 1.1 Adicionar proveedor de datos satisfactoriamente.	Se debe adicionar un nuevo proveedor de datos al sistema.	V E-Prints Complutense	V http://eprints.ucm.es/cgi/oai2	Adiciona el proveedor de datos al sistema y muestra el mensaje: "Se ha creado un nuevo elemento".	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Introduce los datos del nuevo proveedor.

					5. Seleccionar la opción Aceptar.
EC 1.2 Seleccionar la opción Cancelar.	El usuario cancela la opción de adicionar un proveedor de datos.	NA	NA	El sistema muestra un mensaje: "Está seguro que desea cancelar esta operación".	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Introduce los datos del nuevo proveedor. 5. Seleccionar la opción Cancelar. 6. Seleccionar la opción Aceptar.

EC 1.3 Adicionar proveedor de datos con campos vacíos.	Se valida que no existan campos vacíos.	I	V	Impide adicionar el proveedor de datos y muestra el mensaje "Existen campos vacíos, por favor complete estos campos".	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Introduce los datos del nuevo proveedor, dejando algún campo vacío. 5. Selecciona la opción Aceptar.
		V E-Prints Complutens e	I		

<p>EC 1.4 Campos incorrectos.</p>	<p>Se valida que no existan campos incorrectos.</p>	<p>V E-Prints Complutens e</p>	<p>I *&http://eprints.ucm.es/cgi/oai2</p>	<p>Muestra un mensaje de error: "Existen valores incorrectos".</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Introduce los datos del nuevo proveedor, con algún campo incorrecto. 5. Selecciona la opción Aceptar.
<p>EC 1.5 La URL del proveedor de datos ya existe.</p>	<p>Se valida que no se introduzca una URL ya existente.</p>	<p>I E-Prints Complutens e.</p>	<p>V URL ya existente en el sistema.</p>	<p>Muestra el mensaje: "La URL del proveedor de datos ya existe en el sistema".</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar

					<p>la opción Registrar.</p> <p>3. Seleccionar la opción Proveedor de URL.</p> <p>4. Introduce los datos del nuevo proveedor, con una URL ya existente.</p> <p>5. Selecciona la opción Aceptar.</p>
--	--	--	--	--	--

Tabla 13 Caso de Prueba Gestionar proveedores, SC 1 Adicionar proveedor.

Fuente: Elaboración propia

SC 2 Eliminar proveedor

Escenario	Descripción	Nombre	URL	Respuesta del sistema	Flujo central
EC 2.1 Eliminar proveedor de datos.	Se debe eliminar un proveedor de datos del sistema.	NA	NA	<p>Muestra el mensaje: “Ha seleccionado la opción que eliminará los datos introducidos hasta el momento. ¿Desea continuar?”.</p> <p>Una vez aceptada la operación muestra el mensaje: “El elemento se ha</p>	<p>1. Seleccionar en el menú la opción Adquisición.</p> <p>2. Seleccionar la opción Registrar.</p> <p>3. Seleccionar la opción Proveedor de URL.</p>

				eliminado satisfactoriamente”.	<ol style="list-style-type: none"> 4. Introduce los datos del nuevo proveedor. 5. Selecciona la opción Eliminar. 6. Selecciona la opción Aceptar.
EC 2.2 Seleccionar la opción Cancelar.	El administrador cancela la opción de eliminar un proveedor de datos.	NA	NA	<p>Muestra el mensaje “Ha seleccionado la opción que eliminará los datos introducidos hasta el momento. ¿Desea continuar?”.</p> <p>Se cancela la eliminación.</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Introduce los datos del nuevo proveedor. 5. Selecciona la opción Eliminar. 6. Selecciona la opción Cancelar.

Tabla 14 Caso de Prueba Gestionar proveedores, SC 2 Eliminar proveedor.

Fuente: Elaboración propia

SC 3 Modificar proveedor

Escenario	Descripción	Nombre	URL	Respuesta del sistema	Flujo central
EC 3.1 Modificar	Se debe modificar un	I	V	Modifica el proveedor de	1. Seleccionar en el menú la

<p>proveedor de datos.</p>	<p>proveedor de datos del sistema, con los datos correctos.</p>	<p>E-Prints Complutense</p>	<p>http://eprints.ucm.es/cgi/oai2</p>	<p>datos del sistema y muestra el mensaje: “Se ha actualizado los cambios sobre el elemento”.</p>	<p>opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Introduce los datos del nuevo proveedor. 5. Selecciona la opción Modificar. 6. Selecciona la opción Aceptar.</p>
<p>EC 3.2 Seleccionar la opción cancelar.</p>	<p>El usuario cancela la opción de modificar un proveedor de datos.</p>	<p>NA</p>	<p>NA</p>	<p>El sistema muestra la página de lista de proveedores anterior y el proveedor de datos permanece con sus datos originales.</p>	<p>1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Introduce los datos del nuevo proveedor. 5. Selecciona la opción Modificar.</p>

					6. Selecciona la opción Cancelar.
EC 3.3 Modificar proveedor de datos dejando campos vacíos.	Se debe modificar un proveedor de datos del sistema, dejando campos vacíos.	I	V http://eprints.ucm.es/cgi/oai2	Impide modificar el proveedor de datos y muestra el mensaje "Existen campos vacíos, por favor rellene estos campos".	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Introduce los datos del nuevo proveedor. 5. Selecciona la opción Modificar. 6. Selecciona la opción Aceptar.
EC 3.4 Lista de selección vacía.	Se valida que no existan listas de selección vacías.	V	V http://eprints.ucm.es/cgi/oai2	Muestra un mensaje de error: "Seleccione un elemento de la lista".	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL.

					<ol style="list-style-type: none"> 4. Deja la lista vacía. 5. Selecciona la opción Modificar. 6. Selecciona la opción Aceptar.
<p>EC 3.5 Campos incorrectos.</p>	<p>Se valida que no existan campo incorrectos.</p>	<p>V E-Prints Complutense</p>	<p>I *&http://eprints.ucm.es/cgi/oai2</p>	<p>Muestra un mensaje de error: “El campo contiene valores incorrectos”.</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Introduce los datos del nuevo proveedor, con algún campo incorrecto. 5. Selecciona la opción Aceptar.
<p>EC 3.6 La URL del proveedor de datos ya existe.</p>	<p>Se valida que no se introduzca una URL ya existente.</p>	<p>V E-Prints Complutense</p>	<p>V URL ya existente en el sistema.</p>	<p>Muestra el mensaje: “La URL del proveedor de datos ya</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición.

				existe en el sistema".	2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Introduce los datos del nuevo proveedor, con una URL ya existente. 5. Selecciona la opción Modificar. 6. Selecciona la opción Aceptar.
--	--	--	--	------------------------	--

Tabla 15 Caso de Prueba Gestionar proveedores, SC 3 Modificar proveedor.

Fuente: Elaboración propia

SC 4 Listar proveedor

Escenario	Descripción	Nombre	URL	Respuesta del sistema	Flujo central
-----------	-------------	--------	-----	-----------------------	---------------

EC 4.1 Listar los proveedores de datos.	Se debe mostrar una lista con los proveedores de datos.	V E-Prints Complutense	V http://eprints.ucm.es/cgi/oai2	Muestra los datos de los proveedores guardados en el sistema.	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Selecciona la opción Consultar. 5. Selecciona la opción Aceptar.
--	---	------------------------------	--	---	---

Tabla 16 Caso de Prueba Gestionar proveedores, SC 4 Listar proveedor.

Fuente: Elaboración propia

Descripción de las variables

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Nombre	Campo de texto	No	Admite letras, números y caracteres extraños.
2	URL	Campo de texto	No	Admite letras, números y caracteres extraños.

3	Colección	Campo de selección	No	Es donde se seleccionan las colecciones de un proveedor de datos.
4	Servidor	Campo de selección	No	Es donde se seleccionan los proveedor de datos.
5	Fecha de inicio	Campo de selección	No	Es donde se selecciona la fecha de inicio.
6	Fecha de fin	Campo de selección	No	Es donde se selecciona la fecha de fin.

Tabla 17 Descripción de las variables.

Fuente: Elaboración propia

Resultado de las pruebas

A continuación, se muestran los resultados de las pruebas de caja negra que fueron realizadas al componente para el módulo Adquisición, en las que se puede observar el número de iteraciones que se le aplicaron a los casos de pruebas, el total de no conformidades encontradas, a qué estaban asociadas y las que fueron resueltas.

No. de iteraciones	Total de no conformidades	Asociadas a	Resueltas
1ra iteración	10	Errores de ortografía Errores de interfaz	10
2da iteración	3	Errores de ortografía	3
3ra iteración	0	0	0

Tabla 18 Resultados de las pruebas de Caja Negra aplicadas al componente.

Fuente: Elaboración propia

3.5 Conclusiones del capítulo

En el presente capítulo se realizó la descripción de la implementación del componente para el módulo Adquisición del sistema ABCD v3.0 a través del diagrama de despliegue, el cual facilitó mostrar la disposición de las particiones físicas del sistema y la asignación de los componentes de software a estas particiones. Se aplicaron pruebas de Caja Negra utilizando para ello la técnica de partición de equivalencia. Se efectuaron tres iteraciones, encontrando en cada una de ellas varias no conformidades que a su vez fueron corregidas. Se aplicó además la prueba de Funcionalidad, las cuales permitieron comprobar el correcto funcionamiento de los requisitos funcionales del componente y siendo satisfactorios los resultados arrojados.

Conclusiones generales

La presente investigación se centró en el desarrollo de un componente para el módulo Adquisición del Sistema ABCD v3.0 según el estándar OAI-PMH, con la finalidad de facilitar el proceso de recolección de metadatos que se lleva a cabo hoy de forma casi manual en el SIGB. Durante el avance de la misma se cumplieron los objetivos propuestos y se arribó a las siguientes conclusiones:

1. A partir de la elaboración del marco teórico de la investigación se enriqueció el conocimiento acerca de los protocolos de intercambio de información bibliográfica.
2. Se analizó el proceso de interoperabilidad del Sistema ABCD v3.0 con el Sistema REPXOS v3.0.
3. El uso de artefactos ingenieriles sirvió de apoyo en la toma de decisiones con vista a un correcto diseño del componente.
4. Se implementó el componente para el módulo Adquisición del sistema ABCD versión 3.0 a partir de los requerimientos del software establecidos, haciendo uso de la arquitectura y las tecnologías definidas por el equipo de desarrollo de software de dicho sistema.
5. Se diseñaron y aplicaron las pruebas definidas, validando cada una de las funcionalidades del componente para verificar el correcto funcionamiento del mismo.

Recomendaciones

Una vez concluida la investigación se proponen las siguientes recomendaciones:

1. Incluir la funcionalidad Validar servidor con el objetivo de que el usuario pueda verificar el estado del servidor OAI-PMH antes de iniciar el proceso de recolección.
2. Incluir la funcionalidad Seleccionar formato de metadatos para ampliar las posibilidades de recolección del servidor utilizando otros estándares.

Referencias bibliográficas

1. ALVAREZ, L., *Módulo para la prestación de servicios según el estándar OAI-PMH en el SIGB*. S.l.: Universidad de las Ciencias Informáticas. 2013.
2. ANGELFIRE, Patrones GoF. [en línea]. 2011. [Consulta: 25 abril 2018]. Disponible en: <http://geektheplanet.net/5462/patrones-gof.xhtml>.
3. BARRUECO, J.M., OAI-PMH : Protocolo para la transmisión de contenidos en Internet. *Informe APEI sobre acceso abierto* [en línea], 2008. vol. 35, pp. 4. Disponible en: <http://eprints.rclis.org/4093/1/cardedeu.pdf>.
4. BIBLION, 2.12. – La búsqueda y recuperación normalizada de la información bibliográfica: Z39.50 y Z39.50 International Next Generation. *Temario Biblion Ministerio de Cultura Ayudantes A2 Bibliotecas* [en línea]. S.l.: s.n., 2010. pp. 1-7. Disponible en: <http://www.wix.com/bibliotemario/biblion>.
5. BORRERO, M., El sistema integrado abcd en las universidades cubanas. [en línea]. 2013. Disponible en: <http://www.monografias.com/trabajos101/sistema-integrado-abcd-universidades-cubanas/sistema-integrado-abcd-universidades-cubanas.shtml>.
6. BROWN, K. y ARIAS, A., *Diseño e implementación del módulo Catálogo en Línea de Acceso Público del sistema ABCD 3.0*. S.l.: Universidad de las Ciencias Informáticas. 2015.
7. BUSCH, M. y KOCH, N., Rich Internet Applications. *IEEE Internet Computing* [en línea], 2010. vol. 14, no. December, pp. 9-12. ISSN 10897801. DOI 10.1109/MIC.2010.76. Disponible en: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5481362>.
8. CAFFAREL, J., “ Diseño E Implementación De Un Sistema De Control Numerico Por Computadora De Una Fresadora De. , 2017.
9. CAL, W., *plugin* [en línea]. 2015. 2015. S.l.: s.n. Disponible en: <https://docs.supersalud.gov.co/PortalWeb/planeacion/.../GSPD04.docx>.
10. CARPENTIER, L., OAI para principiantes: Introducción. [en línea]. 2003. Disponible en: <http://travesia.mcu.es/portaln/jspui/html/10421/1823/intro.htm>.
11. DEPARTAMENTO DE LENGUAJES Y SISTEMAS INFORMÁTICOS, Tema 3: Patrones de Asignación de Responsabilidades (GRASP). [en línea], 2018. pp. 1-27. Disponible en: www.lsi.us.es/docencia/get.php?id=906.
12. DUEÑAS, L., La Iniciativa de Archivos Abiertos (OAI), un nuevo paradigma en la comunicación científica y el intercambio de información. [en línea], 2005. vol. 1, pp. 36. Disponible en: <http://eprints.rclis.org/9539/1/OAI.pdf>.

13. ECLIPSE, Fundación Eclipse. *9 de abril* [en línea]. 2004. Disponible en: <http://www.eclipse.org/virgo/>.
14. ECLIPSE, Fundación Eclipse. [en línea]. 2015a. Disponible en: <https://www.eclipse.org/swt/>.
15. ECLIPSE, Remote Application Platform. [en línea]. 2015b. Disponible en: <http://www.eclipse.org/rap/>.
16. ECLIPSE FOUNDATION, I., *About the Eclipse Foundation* [en línea]. 2010. 2010. S.l.: s.n. [Consulta: 20 febrero 2018]. Disponible en: <http://www.eclipse.org/org/>.
17. ECLIPSELINK, EclipseLink. [en línea]. 2017. [Consulta: 20 febrero 2018]. Disponible en: <https://eclipse.org/eclipselink/>.
18. GALICIA, Las tecnologías abiertas generan competitividad. [en línea]. 2015. Disponible en: https://www.openfuture.org/es/new/las_tecnologias_abiertas_generan_competitivid.
19. GAVILÁN, C., Selección y adquisición de materiales. Criterios para la formación, mantenimiento y evaluación de la colección bibliográfica. *Temas de Biblioteconomía*, 2008.
20. GÓMEZ, M., Digital libraries: Electronic bibliographic resources on basic education. *Comunicar*, 2012. vol. 20, no. 39, pp. 119-126. ISSN 19883293. DOI 10.3916/C39-2012-03-02.
21. IRLES, A., Sistemas de Gestión de Bases de datos y SIG. *Sistema De Gestión Base De Datos Y Sig* [en línea], 2013. vol. 9, no. 3, pp. 167-180. Disponible en: http://www.um.es/geograf/sigmur/sigpdf/temario_9.pdf.
22. LAFFITA, G., *Universidad de las Ciencias Informáticas Diseño e Implementación del Módulo Adquisición del Sistema ABCD 3.0*. S.l.: s.n. 2015.
23. LARA, M., Pruebas de software. [en línea], 2013. pp. 1-34. [Consulta: 23 abril 2018]. Disponible en: <http://www.pmoinformatica.com/p/pruebas-de-software.html>.
24. LARMAN, C., *Modelo de Casos de Uso: Escritura de Requisitos en Contexto* [en línea]. S.l.: s.n. 2003. Disponible en: <http://www.fmonje.com/UTN/ADES - 208/UML y Patrones 2da Edicion.pdf>.
25. LÓPEZ, P., Práctica 1 Herramienta CASE Visual Paradigm Univ . Cantabria – Fac . de Ciencias Visual Paradigm for UML. , 2017.
26. MARTÍN, C., SIGB Catálogos y gestión de autoridades Diseño y prestaciones de OPACs. *Temas de Biblioteconomía* [en línea], 2008. pp. 1. Disponible en: eprints.rclis.org/13188/1/sigb.pdf.
27. MARTÍN GAVILÁN, C., La biblioteca digital en el ámbito universitario. Integración e interconexión de RE. El movimiento OA y los repositorios de investigación. *Temas de Biblioteconomía (no es revista)*, 2009.
28. MIGUEL ÁNGEL G MEJÍA ARGUETA, La interoperabilidad y el intercambio de metadatos en la red. [en línea], 2013. pp. 1-11. Disponible en: <http://www.revista.unam.mx/vol.13/num9/art95/art95.pdf>.
29. MONTERO, R., OSGI. [en línea], 2016. pp. 1-80. Disponible en:

- http://static1.1.sqspcdn.com/static/f/923743/15025178/1320739253190/OSGI_Roberto_Montero.pdf?token=Eq1hEgb78PYxvuaSJJ35DvCnhDs%3D.
30. MOZILLA DEVELOPER NETWORK, Mozilla Developer Network and individual contributors. [en línea]. 2005. Disponible en: <https://developer.mozilla.org/es/docs/Web/HTML>.
 31. OBS, B.S., ¿Cuál es la utilidad de la matriz de trazabilidad? | OBS Business School. [en línea]. 2018. [Consulta: 11 abril 2018]. Disponible en: <https://www.obs-edu.com/int/blog-project-management/herramientas-esenciales/cual-es-la-utilidad-de-la-matriz-de-trazabilidad>.
 32. PICCO, P. y ORTIZ, V., RDA, el nuevo código de catalogación: cambios y desafíos para su aplicación. *Revista Española de Documentación Científica* [en línea], 2012. vol. 35, no. 1, pp. 146. Disponible en: redc.revistas.csic.es › Inicio › Vol 35, No 1 (2012) › Picco.
 33. PRESSMAN, R.S., *Ingeniería del software. Un enfoque práctico* [en línea]. S.l.: s.n. 2010. ISBN 9786071503145. Disponible en: https://www.tesuva.edu.co/phocadownloadpap/Ingenieria_del_Software._Un_Enfoque_Practico.pdf.
 34. RAMIREZ, L., [Iisis-users] Report on the 2nd online presentation of ISIS & JISIS. [en línea]. 2012. [Consulta: 20 febrero 2018]. Disponible en: <http://lists.iccisis.org/pipermail/isis-users/2012-May/001818.html>.
 35. RÍOS, A.B., *La Estructura Conceptual Del Registro Bibliográfico*. S.l.: s.n. 2003. ISBN 8478006753.
 36. ROJAS, S., Aplicación del esquema de metadatos Dublin Core en la Biblioteca Digital de la Universidad de Antioquia. , 2010. vol. 1, pp. 27.
 37. ROSE, M., What is a widget. [en línea]. 2013. Disponible en: <http://whatis.techtarget.com/definition/widget>.
 38. RUIZ, B. y VAILLANT, C., *Módulo Recomendaciones del sistema para repositorios digitales REPXOS 3.0* [en línea]. S.l.: Universidad de las Ciencias Informáticas. 2015. Disponible en: https://repositorio.uci.cu/jspui/bitstream/123456789/7178/1/TD_07973_15.pdf.
 39. SALINAS, A.G., Z39.50 y OAI-PMH: Protocolos de Transferencia y Recuperación de Información. *XV Conferencia Internacional de Bibliotecología*, 2010. vol. 161.
 40. SAMUEL, qooxdoo Documentation. *Release 6.0.0-alpha* [en línea], 2018. vol. 1, no. 1, pp. 1. Disponible en: <https://media.readthedocs.org/pdf/qooxdoo/latest/qooxdoo.pdf>.
 41. SIZA, P.J., Estudio, análisis, evaluación e implementación de un protocolo de intercambio de contenidos sobre internet para la interoperabilidad de repositorios de la Biblioteca Digital en la Universidad Nacional de Colombia con un proveedor de servicios de contenid. [en línea], 2010. no.

- Maestría thesis, pp. 1-124. Disponible en: <http://www.bdigital.unal.edu.co/1758/>.
42. SOMMERVILLE, I., *Ingeniería del software*. S.l.: s.n. 2011. ISBN 978-607-32-0603-7.
43. UCI, Metodología de desarrollo para la Actividad productiva de la UCI . , 2016. pp. 1-16.
44. UDLAP, Capítulo 3 : Spring, un framework de aplicación. [en línea], 2013. pp. 31. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/sanchez_r_ma/capitulo3.pdf.
45. UNAD, Diagrama de Clases de Diseño | LENGUAJE DE MODELADO UNIFICADO - UML. [en línea]. 2018. [Consulta: 11 abril 2018]. Disponible en: http://stadium.unad.edu.co/ovas/10596_9836/diagrama_de_clases_de_diseo.html.
46. VEL, L.A., UG Lenguaje de Programación : Introducción a C / C ++ (IDE) . , 2010.
47. VELANDIA, M., Como funciona el protocolo OAI-PMH en la recuperación de información. [en línea], 2007. Disponible en: http://eprints.rclis.org/10677/1/COMO_FUNCIONA_EL_PROTOCOLO_OAI_-_PMH_EN_LA_RECUPERACION_DE_INFORMACION.pdf.
48. W3SCHOOLS, XML Introduction. [en línea]. 2015. [Consulta: 20 febrero 2018]. Disponible en: https://www.w3schools.com/xml/xml_what.asp.

Bibliografía

Cmmi. 2010. *Cmmi para desarrollo. Guía para la integración de procesos y la mejora de productos. Tercera edición.* 2010.

Disney, peter. 2006. *Java how to program, seventh edition.* 2006.

Gavilan, cesar martin. 2008. *Sigb catálogos y gestión de autoridades diseño y prestaciones de opacs.* 2008.

Juan, francisco javier martínez. 2011. *Guía de construcción de software en java con patrones de diseño.* 2011.

Larman, craig. 2013. *Uml y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* 2013.

Lawson, bruce. 2011. *Introducing html 5.* 2011.

Llorente, césar de la torre. 2010. *Guía de arquitectura n-capas orientada al dominio.* 2010.

Marquina, ernesto. 2008. *Guía de patrones, prácticas y arquitectura.* 2008.

Miguel, roberto montero. 2013. *Osgi.* 2013.

Pérez, javier eguiluz. 2009. *Introducción a javascript.* 2009.

Pérez, javier eguiluz. 2008. *Introducción a xhtml.* 2008.

Piper, andy. 2011. *Spring dynamic modules in action.* 2011.

Pressman, roger s. 2010. *Ingeniería de software. Un enfoque práctico.* 2010.

Rolando alfredo hernández león, sayda coello gonzález. 2002. *El paradigma cuantitativo de la investigación científica.* Ciudad de la habana : EDUNIV. Editorial universitaria, 2002.

Schildt, Herbert. 2001. *Java 2. Manual de referencia.* 2001.

Glosario de términos del dominio

ABCD: SIGB.

bibliotecario: persona encargada de recolectar la documentación de la biblioteca.

datos_documento: se refiere a los datos de los archivos contenidos en el sistema.

proveedor_datos: sistema donde se encuentra toda la información científica de la UCI.

Anexos

Anexo 1

Las etiquetas de Dublin Core pueden clasificarse en tres categorías (Dublin Core, 2016):

- Sobre el contenido del recurso.

Contenido del recurso *Dublin Core Simple*

Etiquetas <i>DC</i>	Descripción
<i>DC.Title</i>	Título. El nombre dado al recurso bibliográfico.
<i>DC.Subject</i>	Materias y palabras clave. El tema del contenido del recurso bibliográfico.
<i>DC.Description</i>	Descripción. Descripción del contenido del recurso bibliográfico. Puede incluir un resumen, una tabla de contenidos, entre otros.
<i>DC.Source</i>	Fuente. Referencia al recurso del que deriva el documento actual.
<i>DC.Language</i>	Idioma. El idioma del contenido del recurso bibliográfico.
<i>DC.Relation</i>	Relación. Una referencia a un recurso bibliográfico relacionado con el contenido.
<i>DC.Coverage</i>	Cobertura. Ámbito del contenido del recurso bibliográfico. Puede tratarse de una especificación geográfica, temporal o legal.

Tabla 19 Contenido del recurso *Dublin Core Simple*³⁸.

- Sobre la propiedad intelectual del recurso

Propiedad intelectual *Dublin Core Simple*

Etiquetas <i>DC</i>	Descripción
<i>DC.Creator</i>	Autor. Responsable de la creación del contenido. Puede ser una entidad, una persona o un servicio.
<i>DC.Publisher</i>	Editor. Responsable de que el recurso bibliográfico se encuentre disponible.

³⁸ <http://dublincore.org/documents/dces>.

<i>DC.Contributor</i>	Colaborador. Responsable de hacer colaboraciones al contenido del recurso bibliográfico.
<i>DC.Rights</i>	Derechos. Información sobre los derechos de la propiedad intelectual del recurso bibliográfico, como por ejemplo el <i>copyright</i> .

Tabla 20 Propiedad intelectual Dublin Core Simple³⁹.

- Sobre la instancia del recurso.

Instancia *Dublin Core Simple*

Etiquetas DC	Descripción
<i>DC.Date</i>	Fecha. Fecha asociada a la creación o modificación del recurso bibliográfico. Se suele seguir la notación AAAA-MM-DD (año, mes, día respectivamente).
<i>DC.Type</i>	El tipo o categoría del contenido. Palabras claves de un vocabulario que describen la naturaleza del recurso bibliográfico.
<i>DC.Format</i>	Formato. Descripción física del recurso bibliográfico, como su tamaño, duración, dimensiones, entre otros. si son aplicables. Se suelen usar tipos de Extensiones Multipropósito de Correo de Internet (<i>MIME, Multipurpose Internet Mail Extensions</i> , por sus siglas en inglés).
<i>DC.Identifier</i>	Identificación. Referencia unívoca para el contenido del recurso bibliográfico. Por ejemplo una <i>URL</i> o un <i>ISBN</i> .

Tabla 21 Instancia *Dublin Core Simple*⁴⁰.

Anexo 2

Matrices de trazabilidad:

Matriz de trazabilidad MC_Req (Modelo Conceptual_Requerimientos)

³⁹ <http://dublincore.org/documents/dces>.

⁴⁰ <http://dublincore.org/documents/dces>.

(7) Requirement		Adicionar proveedor	Comprobar proveedor de ...	Eliminar proveedor	Listar proveedor	Modificar proveedor	Mostrar colecciones de un ...	Recolectar metadatos por ...
By: Transitor	(4) Class	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	ABCD		<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	bibliotecario		<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	
	datos_documento		<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	proveedor_datos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Ilustración 17 Matriz de trazabilidad MC_Req.

Fuente: Elaboración propia

Matriz de trazabilidad Req_Req (Requerimientos_Requerimientos)

(9) Requirement		Adicionar proveedor	Administrar colección	Comprobar proveedor de ...	Listar proveedor	Controlar proveedores	Eliminar proveedor	Modificar proveedor	Mostrar colecciones de un ...	Recolectar metadatos por ...
By: Transitor	(9) Requirement	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Adicionar proveedor				<input checked="" type="checkbox"/>					
	Administrar colección								<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Comprobar proveedor de ...				<input checked="" type="checkbox"/>					
	Listar proveedor				<input checked="" type="checkbox"/>					
	Controlar proveedores	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
	Eliminar proveedor				<input checked="" type="checkbox"/>					
	Modificar proveedor				<input checked="" type="checkbox"/>					
	Mostrar colecciones de un ...		<input checked="" type="checkbox"/>							
	Recolectar metadatos por ...	<input checked="" type="checkbox"/>								

Ilustración 18 Matriz de trazabilidad Req_Req.

Fuente: Elaboración propia

Matriz de trazabilidad CU_CP (Casos de Uso_Casos de Pruebas)

(4) Test Case		DCP_Comprobar proveedor...	DCP_Gestionar proveedor	DCP_Mostrar colecciones ...	DCP_Recolectar metadato...
By: Transitor					
(4) Use Case		Comprobar proveedor de ...	Gestionar proveedores	Mostrar colecciones de un ...	Recolectar metadatos por ...
		✓			
			✓		
				✓	
					✓

Ilustración 19 Matriz de trazabilidad CU_CP.

Fuente: Elaboración propia

Anexo 3

Acta de aceptación




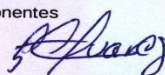

Acta de aceptación

ACTA DE ACEPTACIÓN

En cumplimiento del trabajo de diploma **Componente para Adquisición a través del protocolo OAI-PMH en el sistema ABCD 3.0** se hace entrega de los productos que se relacionan a continuación:

- **Código del componente.**
- **Artefactos generados durante las fases como parte de la metodología AUP.**

El arquitecto del proyecto ABCD y el jefe de Departamento Desarrollo de Componentes, luego de haber revisado los productos de trabajo determina que se **aceptan** y que el mismo cumple con los requisitos definidos y con la calidad necesaria para ser integrada al proyecto.

Entrega	Recibe:
Nombre y apellidos: Maryeris Esperance Aliaga Cargo: Autora del trabajo de diploma Firma: 	Nombre y apellidos: Luis Carlos Álvarez Fernández Cargo: Jefe de Departamento Desarrollo de Componentes Firma: 
	Nombre y apellidos: Alberto A. Aliaga Benítez Cargo: Arquitecto de Proyecto ABCD. Firma: 



Fecha: 12 de junio 2018

Ilustración 20 Acta de aceptación.

Anexo 4

Diseño de casos de pruebas Recolectar metadatos por colección.

Descripción general:

Permite recolectar los datos de una colección dada y especificar rangos de fecha.

Condiciones de ejecución:

El usuario debe estar autenticado en el sistema y poseer el rol de administrador.

SC 1 Recolectar metadatos por colección

Escenario	Descripción	Colección	Servidor	Fecha de inicio	Fecha de fin	Respuesta del sistema	Flujo central
EC 1.1 Recolectar metadatos por colección.	Se realiza la recolección de los metadatos satisfactoriamente.	V Metadata Policy (Política de metadatos).	V E-Prints Complutense	V 10-05-2018	V 10-06-2018	El sistema muestra el mensaje: "Se ha recolectado un nuevo elemento."	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Seleccionar la colección que desea recolectar. 5. Seleccionar los metadatos de la colección que desea recolectar. 6. Selecciona

							la opción Recolectar.
EC 1.2 Campos vacíos.	Se valida que no existan campos obligatorios vacíos.	I	V E-Prints Complutense	V 10-05-2018	V 10-06-2018	Muestra un mensaje de error: "Existen campos vacíos, por favor rellene estos campos".	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Seleccionar la colección que desea recolectar. 5. Seleccionar los metadatos de la colección que desea recolectar. 6. Selecciona
		V Metadata Policy (Política de metadatos).	I	V 10-05-2018	V 10-06-2018		
		V Metadata Policy (Política de metadatos).	V E-Prints Complutense	I	V 10-06-2018		
		V Metadata Policy (Política de metadatos).	V E-Prints Complutense	V 10-05-2018	I		

							la opción Recolectar.
EC 1.3 Rango de fechas incorrecto	Se valida que la fecha de inicio no sea mayor que la fecha de fin.	V Metadata Policy (Política de metadatos).	V E-Prints Complutense	I 10-07-2018	V 10-06-2018	Muestra un mensaje de error: "Rango de fechas incorrecto".	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Seleccionar la colección que desea recolectar. 5. Seleccionar los metadatos de la colección que desea recolectar. 6. Selecciona

							la opción Recolectar.
--	--	--	--	--	--	--	-----------------------

Tabla 22 Caso de Prueba Recolectar metadatos por colección.

Fuente: Elaboración propia

Diseño de casos de pruebas Mostrar colecciones de un repositorio.

Descripción general:

Permite ver las colecciones que posee un repositorio determinado.

Condiciones de ejecución:

El usuario debe estar autenticado en el sistema.

SC 1 Mostrar colecciones de un repositorio

Escenario	Descripción	Colección	Respuesta del sistema	Flujo central
EC 1.1 Mostrar colecciones de un repositorio.	Se muestran las colecciones de un repositorio determinado.	V Identify, ListRecords, ListSets, ListMetadataFormats, ListIdentifiers.	Muestra una lista con las colecciones del repositorio seleccionado.	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Selecciona el repositorio. 5. Selecciona la opción Consultar.
EC 1.2 Colección vacía.	Se muestran las colecciones vacías en un repositorio determinado.	Identify, ListRecords, ListSets, ListMetadataFormats, ListIdentifiers.	Muestra un mensaje: "No se encontraron resultados".	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar.

				<ol style="list-style-type: none"> 3. Seleccionar la opción Proveedor de URL. 4. Selecciona el repositorio. 5. Selecciona la opción Consultar.
EC 1.3 Mostrar colecciones con el campo vacío.	Se valida que el campo no esté vacío para mostrar la colección.		Muestra el mensaje "No ha seleccionado la colección".	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Selecciona el repositorio. 5. Selecciona la opción Consultar.

Tabla 23 Caso de prueba Mostrar colecciones de un repositorio.

Fuente: Elaboración propia

Diseño de casos de pruebas Comprobar proveedor de datos.

Descripción general:

Permite comprobar los datos del repositorio que seleccione.

Condiciones de ejecución:

El usuario debe estar autenticado en el sistema.

SC 1 Comprobar proveedor de datos

Escenario	Descripción	URL	Respuesta del sistema	Flujo central
-----------	-------------	-----	-----------------------	---------------

<p>EC 1.1 Comprobar los datos del repositorio.</p>	<p>Comprueba los datos existentes del repositorio seleccionado.</p>	<p>V http://eprints.ucm.es/cgi/oai2</p>	<p>Muestra un mensaje: "Los datos del repositorio son correctos".</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción Proveedor de URL. 4. Selecciona el repositorio. 5. Selecciona la opción Comprobar proveedor.
<p>EC 1.2 Comprobar los datos de un repositorio que no existe.</p>	<p>Se comprueban los datos de un repositorio que no existe.</p>	<p>V http://eprints.ucm.es/cgi/oai2</p>	<p>Muestra un mensaje: "No se encontraron resultados".</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición. 2. Seleccionar la opción Registrar. 3. Seleccionar la opción proveedor de URL. 4. Seleccionar el repositorio. 5. Seleccionar la opción Comprobar.
<p>EC 1.3 Comprobar proveedor con el</p>	<p>Se valida que el campo no esté vacío para comprobar el</p>	<p>I</p>	<p>Muestra el mensaje: "El campo URL está vacío, por rellene</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú la opción Adquisición.

campo vacío.	proveedor.		el campo”.	<ol style="list-style-type: none">2. Seleccionar la opción Registrar.3. Seleccionar la opción Proveedor de URL.4. Selecciona el repositorio.5. Selecciona la opción Comprobar.
--------------	------------	--	------------	---

Tabla 24 Caso de Prueba Comprobar proveedor de datos.

Fuente: Elaboración propia