

UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS

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

**Módulo de recuperación de metadatos para la Plataforma de
recursos de la empresa Editorial Poligráfica *Félix Varela***

Autor:

Alex Acosta Montejo

Tutores:

Ing. Yordankis Matos López

Ing. Elias Bello Camps

Ing. Yenisel Díaz Llanes

La Habana, junio de 2018

Declaración de Autoría

Declaro ser autor del presente trabajo de diploma para optar por el título de Ingeniero en Ciencias informáticas y reconozco a la Universidad los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los __ días del mes de junio del año 2018.

Firma del autor:

Alex Acosta Montejo

Firma de los tutores:

Ing. Yordankis Matos López

Ing. Elias Bello Camps

Ing. Yenisel Díaz Llanes

Datos de contacto

Ing. Elias Bello Campos: Graduado en el año 2015 como Ingeniero en la Universidad de Ciencias Informáticas.

Correo electrónico: ebcamps@uci.cu

Ing. Yenisel Díaz Llanes: Graduada en el año 2016 como Ingeniero en la Universidad de Ciencias Informáticas.

Correo electrónico: ydiaz@uci.cu

Ing. Yordankis Matos López: Graduado en el año 2008 como Ingeniero en la Universidad de Ciencias Informáticas.

Correo electrónico: yluquen@uci.cu

Agradecimientos

A mi familia por su cariño, en especial a mi madre, mi padre y mi hermano, por ser los principales educadores de mi vida.

A Raymari por ser mi compañera y amiga en estos 5 años de universidad.

A mis fieles amigos, que siempre me tuvieron presente y se mantuvieron a mi lado dándome ánimos para alcanzar la meta.

A los profesores, que me dedicaron parte de su tiempo para ayudarme y aconsejarme.

Y a todos aquellos que se preocuparon por cómo marchaba mi investigación.

Dedicatoria

A mis padres porque sin ellos nada de esto hubiera sido posible si no fuera por su dedicación e interés por mi superación y por todo el sacrificio que han tenido que hacer para que hoy finalmente pudiera ser Ingeniero.

A mi hermano por ser mi ejemplo a seguir y el espejo en el que siempre me he mirado.

La Editorial Poligráfica *Félix Varela* (EPPFV) es la encargada de la realización del proceso de edición de libros académicos que se corresponden con los programas de estudio de las carreras del Sistema de Educación Superior Cubano. Esta cuenta actualmente con una plataforma para la gestión de los servicios entre los cuales se encuentra la publicación y promoción online de libros y recursos educativos, pero no brinda la posibilidad de recuperar información de otros repositorios digitales. La presente investigación está dirigida a desarrollar un módulo que le permita recuperar desde la plataforma metadatos disponibles en otros sistemas. Para el desarrollo de la misma, se realizó un estudio sobre los protocolos de intercambio y recuperación de información donde se selecciona el estándar OAI-PMH para recuperar metadatos almacenados en otros repositorios digitales y de esta forma darle respuesta al problema que dio origen a esta investigación. Este documento describe el desarrollo de dicho módulo, que proveerá los servicios requeridos, fue desarrollado mediante la metodología AUP, se utiliza el lenguaje de programación PHP 5.4 y el framework de desarrollo Symfony en su versión 2.3.7. Se realizaron los flujos de Modelado, Implementación y Prueba, indicados por la metodología elegida, obteniéndose un sistema que cumple con los requerimientos funcionales y no funcionales. Con el módulo desarrollado, se permite la recolección de metadatos de otros repositorios digitales, facilitando un acceso a estos de forma, rápida y sencilla.

Palabras clave: estándar, OAI-PMH, protocolos, repositorio digital.

Índice de Contenidos

Capítulo 1. Fundamentación teórica.....	5
1.1 Conceptos y definiciones asociados al problema de investigación	5
1.2 Protocolos para la comunicación entre sistemas. Definición y características	5
1.3 Protocolos de intercambio y recuperación de la información.....	7
1.4 Flujo actual de los procesos de OAI-PMH	9
1.5 Análisis de soluciones similares	12
1.6 Metodología de desarrollo, herramientas y lenguajes.	14
1.7 Conclusiones parciales	19
Capítulo 2. Propuesta de solución	20
2.1 Descripción de la propuesta de solución.....	20
2.2 Modelo de dominio.....	20
2.3 Especificación de requisitos de software	21
2.4 Historias de usuarios	24
2.5 Patrón Arquitectónico	26
2.6 Patrones de diseño	27
2.7 Modelo de diseño.....	28
2.8 Diagrama de despliegue	29
2.9 Modelo de datos.....	30
2.10 Conclusiones parciales	33
Capítulo 3: Implementación y PRUEBAS	34
3.1 Modelo de implementación.....	34
3.3 Pruebas de software.....	38
3.4 Niveles de pruebas.....	39
3.5 Resultados obtenidos en las pruebas.....	46
3.6 Conclusiones parciales.....	48
Conclusiones	49
Recomendaciones	50
Bibliografía	52
Anexos.....	56

Índice de Tablas

Tabla 1: Requisitos Funcionales	21
Tabla 2: HU del requisito Adicionar repositorios OAI-PMH.....	24
Tabla 3: Descripción de la tabla serveroaimh.....	30
Tabla 4: Descripción de la tabla metadata-recolected.....	31
Tabla 5: Caso de prueba perteneciente al RF Adicionar repositorio OAI-PMH.....	42
Tabla 6: Resultados de las pruebas del sistema	46
Tabla 7: Clasificación de las No Conformidades por el tipo de error	46
Tabla 8: HU Modificar repositorios OAI-PMH	56
Tabla 9: HU Eliminar repositorios OAI-PMH.....	58
Tabla 10: HU Listar repositorios OAI-PMH.....	60
Tabla 11: HU Buscar documentos indexados	61
Tabla 12: HU Recolectar metadatos	64
Tabla 13: Caso de prueba perteneciente al RF Buscar documentos indexados.....	66
Tabla 14: Caso de prueba perteneciente al RF Mostrar los resultados de la búsqueda	68
Tabla 15: caso de prueba perteneciente al RF Listar repositorios OAI-PMH.....	70
Tabla 16: Caso de prueba perteneciente al RF Eliminar repositorio OAI-PMH.....	72
Tabla 17: Caso de prueba perteneciente al RF Modificar repositorio OAI-PMH	75

Índice de Figuras

Fig. 1: Diagrama del modelo de dominio.....	20
Fig. 2 Diagrama de clases .Elaboración propia.....	29
Fig. 3 Diagrama de despliegue. Elaboración propia.....	30
Fig. 4 Diagrama entidad-relación. Elaboración propia.....	30
Fig. 5 Formulario Crear Servidor OAI-PMH.....	35
Fig. 6 Búsqueda de metadatos	36
Fig. 7 Detalles del metadato	37
Fig. 8 Diagrama de Componentes. Elaboración propia	38
Fig. 9 No conformidades detectadas.....	47
Fig. 10 Diagrama de clases Mostrar y Buscar metadatos	79

Introducción

En el mundo actual, gracias al desarrollo de las Tecnologías de la Información y la Comunicación TIC, se ha logrado almacenar, compartir y gestionar grandes colecciones de datos en formato digital. Con el uso de esta tecnología la producción de información ha ido creciendo de forma acelerada, la cual se encuentra disponible en diferentes formatos y con diversos propósitos como investigación y enseñanza. El acceso a la información es un proceso que involucra a investigadores, editores, bibliotecarios, sociedades científicas y universidades.

En Cuba, existe un crecimiento acelerado de las TIC, por lo que la comunidad tiene acceso a gran parte de la información de contenido digital, la cual impulsa el desarrollo educativo y profesional de estudiantes y profesores de diferentes carreras universitarias. Debido a lo antes mencionado se decide desarrollar como alternativa para acceder a la información, diferentes plataformas web para facilitar el acceso a libros y a publicaciones científicas y literarias.

En el Centro de Tecnologías para la Formación (FORTES) de la Facultad 4 de la Universidad de las Ciencias Informáticas (UCI), se desarrolló el proyecto: Plataforma de recursos para la Empresa Editorial Poligráfica *Félix Varela* (PREEPFV), la cual es la encargada de la realización del proceso de edición de libros académicos que se corresponden con los programas de estudio de las carreras del Sistema de Educación Superior Cubano. Sus publicaciones constituyen un vasto espacio de consulta y localización de materiales tanto para estudiantes como profesores.

El proyecto ha desarrollado una plataforma para gestionar cada uno de sus servicios, siendo estos la publicación y promoción online de libros y recursos educativos con funcionalidades como: búsqueda sencilla, búsqueda avanzada y exploración que facilitan al usuario la localización de los documentos con base en los metadatos.

Los metadatos son datos, información, que dotan a un documento de contenido, contexto y estructura. Esta definición, sin duda, va más allá de decir que son “datos sobre los datos”. Los documentos digitales están dotados de metadatos que los identifican, los ubican en un contexto de proceso y temporal, los describen y los hacen únicos[1]. El principal valor agregado de un sistema como la Empresa Editorial Poligráfica Félix Varela consiste en garantizar la disponibilidad de la información en todo momento. Sin embargo, esta última no es capaz de recuperar información sobre el material custodiado por otros repositorios digitales, aún y cuando estos utilizan un sistema informático con mecanismos para la consulta del mismo.

Introducción

La imposibilidad de acceder a la información almacenada en otros repositorios digitales impide una mejor difusión de la memoria histórica custodiada por las mismas. Además, no permite el acceso a materiales vitales para el desarrollo investigativo por parte de los usuarios finales de esta información que son mayormente estudiantes y profesores.

Por lo anteriormente expuesto se plantea como **problema a resolver**: ¿Cómo recuperar desde la Plataforma de recursos de la Empresa Editorial Poligráfica Félix Varela, metadatos almacenados por repositorios digitales?

Por la problemática antes descrita se define como **objeto de estudio**: La recuperación de metadatos. Enfocando el **campo de acción en** La recuperación de metadatos almacenados en repositorios digitales. Teniendo en cuenta el problema a resolver se propone como **objetivo general**: desarrollar un módulo para la Plataforma de recursos de la Empresa Editorial Poligráfica Félix Varela que permita a dicho sistema recuperar metadatos almacenados en otros repositorios digitales. El presente trabajo queda sustentado por la **idea a defender**: el desarrollo de un módulo para la recuperación de metadatos permitirá a la Plataforma de recursos de la editorial Félix Varela, proveer a sus usuarios de información sobre los materiales almacenados en otros repositorios digitales.

A partir del objetivo general definido se derivan los siguientes **objetivos específicos**:

1. Construir el marco teórico referencial mediante la consulta, extracción y recopilación de información relevante sobre la recuperación de metadatos.
2. Desarrollar el análisis y diseño del módulo de recuperación de metadatos para la Plataforma de recursos de la Editorial Félix Varela.
3. Implementar un módulo para la Plataforma de recursos de la Empresa Editorial Poligráfica Félix Varela.
4. Validar el correcto funcionamiento de la propuesta de solución con respecto a los requisitos definidos al inicio del desarrollo.

Posibles resultados:

Módulo capaz de recuperar metadatos de otros repositorios digitales, además de permitir el acceso en línea a materiales digitales.

Métodos de investigación:

Métodos teóricos.

- **Analítico sintético:** utilizado para identificar las distintas partes que intervienen en el problema, analizarlas por separado y luego integrarlas para obtener una visión global del mismo.
- **Histórico lógico:** empleado para analizar la evolución de los protocolos de intercambio de datos entre sistemas de información, su condicionamiento a los diferentes períodos de la historia, así como el análisis de los principales Sistemas de Gestión de Documentos que pueden intercambiar información entre ellos.
- **Modelación:** utilizado en el esbozo de los diferentes diagramas y modelos generados durante el proceso de desarrollo de un módulo que contribuya a la recuperación de metadatos.

Métodos Empíricos.

- **Observación:** permite estudiar de cerca el objeto de la investigación, las acciones, causas y consecuencias logrando conocer la esencia del problema planteado, analizando desde varios puntos de vista la propuesta de solución y otras soluciones existentes e identificando que está hecho y que falta por hacer.

El presente trabajo consta de una introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y anexos, el cual constituye una constancia del modo en el que se da cumplimiento a los objetivos planteados. Los capítulos están estructurados de la siguiente manera:

Capítulo 1. Fundamentación teórica:

Se abordarán los conceptos fundamentales a tener en cuenta para comprender los términos usados en el desarrollo del trabajo. Se realiza un estudio de los principales protocolos para lograr la interoperabilidad entre Sistemas de Gestión de Documentos. Además, se exponen las distintas tecnologías a utilizar en el desarrollo del módulo propuesto.

Capítulo II: Propuesta de solución:

Introducción

Comprende los requisitos funcionales y no funcionales del módulo a desarrollar. En este capítulo se especifica además el diagnóstico del campo de acción y sus particularidades con el objetivo de realizar una propuesta de solución a partir de lo analizado en el capítulo anterior.

Capítulo III: Implementación y pruebas:

Contempla la implementación de la propuesta de solución teniendo en cuenta el alcance especificado para el trabajo de diploma, así como la verificación de la calidad de software a través de aceptación y unitarias x caja negra

Capítulo 1. Fundamentación teórica

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se presentan los conceptos fundamentales para comprender los términos usados en la presente investigación. Se realiza un estudio de los principales protocolos para lograr la interoperabilidad entre Sistemas de Gestión de Documentos. Además, se realiza una breve descripción de las distintas herramientas y tecnologías que serán utilizadas en el desarrollo del módulo.

1.1 Conceptos y definiciones asociados al problema de investigación

Los archivos generalmente son conocidos como los edificios que contienen los documentos o el mueble que los guarda. Sin embargo, este término presenta tres significados: como contenido documental, como institución y como lugar de conservación. El diccionario de terminología archivística, lo define con tres acepciones [2]:

- Conjunto orgánico de documentos producidos y/o recibidos en el ejercicio de sus funciones por las personas físicas o jurídicas, públicas y privadas.
- La institución cultural donde se reúnen, conservan, ordenan y difunden los conjuntos orgánicos de documentos para la gestión administrativa, la información, la investigación y la cultura.
- El archivo también es el local donde se conservan y consultan los conjuntos orgánicos de documentos.

Estos archivos conforman lo que es un repositorio digital, siendo este una colección de objetos digitales organizada, que sirve a una comunidad de usuarios definida, que tiene los derechos de autor presentes y gestionados y que dispone de mecanismos de preservación y conservación. Es un sistema de tratamiento técnico, acceso y transferencia de información digital, estructurado alrededor del ciclo de vida de una colección de documentos digitales, sobre los cuales se ofrecen servicios interactivos de valor añadido para el usuario final [3].

1.2 Protocolos para la comunicación entre sistemas. Definición y características

En informática, un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red, es una convención o estándar que controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales [4].

Capítulo 1. Fundamentación teórica

Un protocolo es clave para lograr la interoperabilidad, la cual 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” [5].

Un sistema interoperable garantiza el flujo de metadatos, los cuales deben su surgimiento a la enorme cantidad y variedad de las fuentes y recursos de información disponibles en Internet, surgió así el problema de clasificarla e identificarla de manera rápida y eficiente por lo que se hizo necesario crear un mecanismo que fuese capaz de etiquetar, catalogar, describir y clasificar los recursos presentes en la Web con el fin de facilitar la búsqueda y recuperación de la información. Este mecanismo es a lo que hoy se le conoce con el nombre de metadatos.

Los metadatos son la información estructurada acerca de recursos (digitales y no digitales) [6]. Uno de los objetivos que presentan los metadatos es ayudar a darle soporte y publicidad a los datos que una persona u organización han producido.

Existen diversas clasificaciones de los metadatos atendiendo a distintos aspectos como su forma, funcionalidad, nivel de estructuración de los datos, persona o entidad que los origina, etc. La calificación que se ofrece a continuación es según la función que proporcionan [7]:

- **Metadatos descriptivos:** son aquellos que sirven para la descripción e identificación de los recursos de información, permiten la búsqueda y recuperación de la información, como también distinguir un recurso de otro y entender el asunto o contenido del mismo.
- **Metadatos estructurales:** son los que más influyen en la recuperación de la información electrónica, facilitan la navegación y presentación de los recursos electrónicos. Ofrecen la información sobre la estructura interna de los recursos, estableciendo las relaciones entre ellos, de manera que pueden incluso unir los archivos de imagen y textos que están relacionados.
- **Metadatos administrativos:** son de carácter más técnico porque incluyen datos sobre la creación y control de calidad, datos sobre la gestión de derechos, requisitos del control de acceso y utilización, información sobre la preservación y permiten la gestión a largo y corto plazo.

Capítulo 1. Fundamentación teórica

1.3 Protocolos de intercambio y recuperación de la información

Cualquier desarrollo informático que involucre un procesamiento de datos e información, y que apoye a las organizaciones en la optimización, mejoramiento y desarrollo de sus procesos puede ser considerado un sistema de información. Los Sistemas de Gestión de Documentos corresponden a un tipo especial de sistema de información. A partir de la evolución de las TIC, surge la necesidad de establecer la interoperabilidad entre los sistemas de información, y por ende entre los Sistemas de Gestión de Documentos, con el objetivo de posibilitar la transferencia e intercambio de información entre ellos [5]. Para que esta comunicación pueda establecerse de una manera organizada surgieron algunos protocolos dentro de los cuales se pueden encontrar:

➤ **Protocolo Dienst**

Permite la comunicación entre servicios que conforman una biblioteca digital. Fue diseñado y probado con el sistema NCSTRL (*Networked Computer Science Technical Report Library*) [8]. Parte de cuatro tareas básicas: búsqueda y recuperación de documentos, navegación, agregación de nuevos documentos y registro de usuarios, y propone seis categorías de servicios: repositorio, indexado, mediadores, información, colección y registro. En la actualidad este protocolo es muy poco usado por las instituciones.

➤ **Protocolo Guildford**

Proporciona un conjunto de reglas para la publicación y el intercambio de documentos en el Internet, este puede ser implementado tanto individualmente como en grupos y es especialmente adecuado si los metadatos se codifican en REDIF (por su siglas en inglés, *Research Document Information Format*) [9]. Su surgimiento se remonta a una declaración realizada por William L. Goffe el 15 de julio de 1995, donde expresó la necesidad que existía de un sistema distribuido con cualquier número de sitios, donde cada uno fuera el reflejo del otro. Este protocolo fue creado con el objetivo de facilitar el acceso a los últimos resultados de investigaciones en las diferentes áreas de la economía, lo cual no quiere decir que no pueda ser aplicable a otros sectores de la sociedad [9].

➤ **Protocolo Simple para la Interoperabilidad de Bibliotecas Digitales (SDLIP)**

SDLIP es un protocolo para la integración de múltiples fuentes, de información heterogénea. Ha sido desarrollado conjuntamente por la Universidad de Stanford, UC Berkeley, UC Santa Bárbara, el San Diego Supercomputer Center, y el Proyecto de Biblioteca Digital de California. Los clientes utilizan SDLIP para solicitar realizar búsquedas sobre las fuentes de información [10].

Capítulo 1. Fundamentación teórica

➤ **Protocolo Z39.50**

Z39.50 es un protocolo para la recuperación de información basado en la estructura cliente/servidor que facilita la interconexión de sistemas informáticos. Es un protocolo desarrollado y mantenido por bibliotecarios y se adopta como la norma ISO 23950. Su principal objetivo consiste en permitirle al usuario realizar búsquedas en bases de datos que cuenten con un servidor Z39.50, sin tener que conocer las sintaxis de búsqueda que utilizan dichos sistemas. La funcionalidad que realiza este protocolo es muy compleja, por ejemplo, trata cuestiones como el manejo de sesiones, gestión de conjuntos de resultados y permite la especificación de predicados para filtrar los resultados obtenidos, proporcionando así un incremento en la complejidad de la implementación y por ende un aumento de los costos. Este protocolo ha sido utilizado mayoritariamente por bibliotecarios y a partir del empleo de estándares como el formato MARC [11].

Iniciativa de Archivos Abiertos –Protocolo para la Recolección de Metadatos (OAI-PMH)

El marco diseñado por OAI (por sus siglas en inglés, *Open Archives Initiative* o Iniciativa de Archivos Abiertos) es bastante simple y posee como objetivo una mínima complicación para las instituciones que en algún momento deseen implementarlo. Si bien OAI-PMH surgió en el seno de la comunidad académica y científica para la búsqueda y recuperación de textos electrónicos, es perfectamente aplicable en cualquier contexto documental [12].

Después de haber realizado un estudio de algunos de los protocolos para lograr interoperabilidad entre sistemas, se decide emplear OAI-PMH para la de metadatos. La selección de este protocolo estuvo basada en los siguientes aspectos [12]:

- Simplicidad a la hora de implementar.
- Cantidad de instituciones de archivo que lo han desarrollado.
- Presenta mayor interoperabilidad que otros protocolos pues a través de uno de los verbos del protocolo es posible obtener los formatos que implementa el proveedor de datos, así como realizarle peticiones en cualquiera de ellos.
- Está basado en estándares ampliamente utilizados, como son el HTTP para la transmisión de datos y órdenes y XML (Lenguaje de Marcas Extensible) para la codificación de metadatos.

Capítulo 1. Fundamentación teórica

1.4 Flujo actual de los procesos de OAI-PMH

OAI-PMH no es un protocolo de búsqueda, pero su utilización puede servir de apoyo para los servicios de búsqueda, es una capa sobre la que construir otros servicios. Este protocolo utiliza transacciones HTTP para emitir preguntas y obtener respuestas entre un proveedor de servicios (PS) y un proveedor de datos (PD). El primero puede pedir al segundo que le envíe metadatos según determinados criterios. En respuesta, el segundo devuelve un conjunto de registros en formato XML, incluyendo identificadores de los objetos descritos en cada registro [13].

Las peticiones se realizan utilizando los métodos GET o POST del protocolo HTTP y constan de una lista de opciones con la forma de pares del tipo clave=valor. Existen seis tipos de peticiones que un proveedor de servicio puede realizar a un proveedor de datos, las cuales se conocen como verbos [14]:

Identify

Esta petición se utiliza para que el PS tenga conocimiento de la información tanto técnica como administrativa del PD con el objetivo de establecer cómo se va a realizar el proceso de recolección de metadatos. La información remitida por el PD al PS debe incluir una instancia de los siguientes elementos:

- Nombre del repositorio.
- Base URL.
- Versión del protocolo OAI-PMH por el cual se va a realizar el cosechado.
- earliestDatestamp: expresa la fecha del último cambio de adición, eliminación o modificación de los objetos dentro del repositorio.
- deletedRecord: establece la forma como se efectúa el proceso de eliminación de los objetos dentro del repositorio, que puede ser de tres tipos: no se realiza, transitoria o persiste.
- Granularidad: define el formato de espacio y tiempo bajo el cual está soportado el repositorio, tiene que estar definida en la norma ISO 8601 31.
- Correo electrónico del administrador del repositorio.

ListMetadataFormats

Esta petición se realiza para conocer bajo qué formatos de metadatos el PS puede recuperar la información dada por el PD, por lo general este protocolo se basa en los metadatos Dublin Core que será descrito más

Capítulo 1. Fundamentación teórica

adelante. Con los resultados de esta petición, el PS elegirá la sintaxis de metadatos que utilizará para recuperar la información del PD y se convertirá en un argumento `metadataPrefix`, con el cual el PD remitirá los objetos resultantes para la posterior cosecha.

ListSets

Esta petición se utiliza para recuperar un conjunto de registros. Estos conjuntos son creados de forma opcional por el servidor para proveer una recuperación selectiva de los registros. Estos conjuntos pueden ser simples listas o estructuras jerárquicas.

ListIdentifiers

Esta petición se utiliza para obtener la información del verbo `ListRecords` de una forma más resumida, pero con la diferencia de que aquí solamente se recuperarán los encabezamientos de los registros, en lugar de los registros completos, por esta razón es posible afirmar que los argumentos requeridos por este verbo son iguales a los que se solicitan cuando se hace una petición del verbo `ListRecords`.

Con estas cabeceras, el PS obtiene más información sobre el identificador OAI del objeto a recolectar, el `timestamp` o fecha de ingreso o modificación del objeto dentro del sistema y los `setSpec` o identificadores del set relacionados, que son por los cuales se agrupan los objetos del repositorio.

ListRecords

Esta petición es la que se utiliza para recolectar los objetos que están dentro de un repositorio, con el fin de darle un resultado al PS relacionado con las necesidades del mismo; este verbo puede tener como argumentos las siguientes variables:

- Fechas (superior e inferior) dentro de las cuales se pueden agrupar los metadatos de los objetos a cosechar.
- Set: es la agrupación por alguna característica descriptiva de los contenidos dentro del repositorio; si no se envía el set o los setters dentro de los argumentos del verbo `ListRecords`, entonces el resultado de esta petición serán todos los objetos que se encuentran dentro del repositorio, sin excepción alguna.

Capítulo 1. Fundamentación teórica

- ResumptionToken: es un argumento exclusivo que se maneja dentro del control de intercambio de información de este protocolo. Es importante destacar que se utiliza cuando la lista de metadatos de objetos del repositorio es muy grande, cuando esto sucede se realiza un envío por lotes comunicándole al PS cuántos son, cuántos se han transmitido y si desea cosechar todos los metadatos asociados a esta petición.

GetRecord

Esta petición se utiliza para recuperar un registro concreto, para lograr esto, se le debe realizar la petición al PD con los siguientes argumentos:

- Identifier: es la llave o número único dentro del repositorio que identifica al objeto dentro del sistema.
- MetadataPrefix: es la sintaxis de metadatos que se va a utilizar para realizar el cosechado del objeto.

OAI-PMH soporta múltiples formatos para expresar los metadatos, no obstante, requiere que todos los servidores ofrezcan los registros utilizando Dublin Core codificado en XML [12]. Además, cabe destacar que cada servidor puede ofrecer los registros en otros formatos adicionales. Un cliente puede solicitar que los registros se le brinden en cualquier otro formato que sea soportado por el servidor.

Dublin Core es un modelo de metadatos elaborado por la DCMI (*Dublin Core Metadata Initiative*), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos y a promover el desarrollo de los vocabularios especializados de metadatos para describir recursos. Las implementaciones de Dublin Core usan generalmente XML y se basan en el *Resource Description Framework* [15].

Dublin Core tiene como objetivo otorgarle un significado semántico a las quince definiciones descriptivas que presenta. Cada definición es opcional y puede repetirse. Además, estas pueden aparecer en cualquier orden. Este sistema de definiciones fue diseñado especialmente para proporcionar un vocabulario de características base, que fuese capaz de facilitar la información descriptiva básica referente a cualquier recurso, sin que importe cual es el formato de origen, el área de especialización o el origen cultural [15].

Capítulo 1. Fundamentación teórica

El protocolo OAI-PMH se basa en el modelo cliente/servidor que transmiten preguntas y respuestas entre un PD y un PS, los primeros son los archivos que suministran la información y los segundos son los recolectores que toman los datos, con el objetivo de incorporarles algún valor añadido y presentarlos a los usuarios finales [16].

Cuando un proveedor de datos recibe una petición de metadatos, este primeramente la procesa a través de una capa de aplicación programada, la cual contiene una implementación de OAI-PMH para realizar una consulta a su sistema de gestión de base de datos que contiene los metadatos relativos a los recursos que se encuentran en el repositorio. Una vez hecha la consulta la aplicación debe haber recopilado los metadatos, compone la respuesta a la petición realizada en formato OAI-PMH con sintaxis XML y posteriormente se envía a través de HTTP [16].

Luego la capa de aplicación del proveedor de servicio recolecta la respuesta y se encarga de introducir en su propia base de datos los metadatos recibidos para componer a continuación los servicios de valor añadido que presentará al usuario final [16].

Los proveedores de datos manejan el depósito y la publicación de los recursos en un repositorio y exponen los metadatos de los recursos del repositorio para que puedan ser recolectados. Ellos son los creadores y conservadores de los metadatos y de los repositorios de recursos [13].

Los PS recolectan los metadatos de los PD. Ellos emplean los metadatos recolectados con el fin de proporcionar servicios [13].

1.5 Análisis de soluciones similares

El auge tecnológico ha traído consigo un mayor avance en los sistemas de gestión de archivos históricos, existiendo en la actualidad varios *software* que le permiten a estos difundir sus fondos en internet. Esto se puede lograr implementando un protocolo para el intercambio y recuperación de metadatos, siendo el más adecuado para la gestión archivística OAI-PMH. A continuación, se muestran algunas aplicaciones existentes que implementan este protocolo (Dspace, EPrints) con el objetivo de determinar si su código fuente puede ser reutilizado para la implementación del módulo a desarrollar.

Capítulo 1. Fundamentación teórica

DSpace

Es un repositorio de documentos digitales con preservación a largo plazo, es un archivo estandarizado de documentos con un sistema de indexación y búsqueda sobre metadatos y en el texto completo (Opcional). Dspace crea URL's permanentes para los materiales almacenados y permite la realización de copias de seguridad automáticamente de los archivos de una institución a otra. Dspace es un sistema de información con arquitectura de repositorio digital que captura, almacena, ordena, preserva y distribuye material de investigación digital con el propósito de garantizar que se preserve y distribuya toda la producción intelectual generado al interior de las instituciones que hacen uso de este [17].

Además del código basado en los estándares, la arquitectura también se base en estándares como Dublín Core, METS, OAI, CNI Handles para la persistencia de los URL, y otros. Otra ventaja técnica es la utilización del motor de bases de datos Postgresql el cual es poderoso y robusto [17].

EPrints

Es un repositorio de documentos digitales que permite la publicación de distintos objetos (variedad de contenido) por ejemplo: libros, conferencias, objetos de aprendizaje, multimedia, publicaciones, referencias, entre otros, además, de permitir la obtención de un elemento cargado mediante un motor de búsqueda en el cual se pueda buscar por la mayor cantidad de criterios posibles. Otra característica fundamental es que los metadatos pueden ser exportados en diferentes formatos para poder realizar migraciones de versión del mismo producto y migraciones a otros productos. Se basa en el protocolo OAI-OMH, utilizando MySQL como gestor de base de datos y Perl como lenguaje de desarrollo.[18].

Con el estudio realizado se han identificados las siguientes funcionalidades como semejanzas entre las mismas que servirán de guía a la propuesta a desarrollar:

- Permiten realizar búsquedas sencillas y avanzadas.
- Posibilitan la gestión y difusión de archivos.
- Pueden interoperar con otros sistemas.
- Permiten la recuperación de información de registros en diferentes repositorios gracias a OAI/PMH.
- Utilizan como formato de metadato Dublin Core.

Capítulo 1. Fundamentación teórica

Sin embargo, existen características fundamentales que impiden la reutilización del código de los sistemas antes mencionado, siendo la principal, que ninguna de las 2 pueden ser integradas a la Plataforma debido al lenguaje de programación en el que se encuentran desarrolladas.

1.6 Metodología de desarrollo, herramientas y lenguajes.

La arquitectura establecida para la PREEPFV define como metodología de desarrollo de software Variación de AUP para la UCI en su escenario cuatro, además establece herramienta CASE, lenguaje de modelado, lenguaje del lado del cliente, lenguaje del lado del servidor, gestor de base de datos, framework e IDE de desarrollo y servidor web, los cuales serán descritos a continuación.

Metodología de desarrollo de software

Una metodología de desarrollo de software es un conjunto de métodos, estas se elaboran a partir del marco definido por uno o varios modelos del ciclo de vida. Esta consiste en un conjunto de procedimientos, técnicas, herramientas y soporte documental, que deben seguirse para el desarrollo de un software. Especifican [19] :

- La división de un proyecto en etapas.
- Las tareas que hay que realizar en cada etapa.
- Las salidas que se producen y cuándo.
- Las herramientas que se utilizan.
- La gestión y el control de un proyecto.

Variación de AUP para la UCI

La metodología AUP en su versión UCI define tres fases: Inicio, Ejecución y Cierre. A continuación, se describe cada una de ellas [20]:

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.

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

Capítulo 1. Fundamentación teórica

modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

Cierre: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Para el Modelado del dominio se plantean tres formas de encapsular los requisitos: Casos de uso del sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por procesos (DRP), de los cuales surgen cuatro escenarios para modelar el sistema en los proyectos quedando de la siguiente forma [20]:

Escenario No 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema. Es necesario que se tenga claro por el proyecto que los CUN muestran como los procesos son llevados a cabo por personas y los activos de la organización.

Escenario No 2: 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.

Escenario No 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. Se debe tener presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados.

Escenario No 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una HU no debe poseer demasiada información.

Lenguaje Unificado de Modelado (UML) 2.1

Capítulo 1. Fundamentación teórica

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.[21]

Herramienta CASE

Visual Paradigm for UML 8.0 Enterprise Edition, es una herramienta profesional multiplataforma y con licencia gratuita que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda una rápida construcción de aplicaciones de calidad y a un menor coste. Permite crear todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. [22]

Lenguajes utilizados del lado del cliente:

HTML5

HTML, siglas de HyperText Markup Language («lenguaje de marcado hipertextual»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir o hacer referencia a un tipo de programa llamado script, el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML [23].

CSS3

Cascading Style Sheets u Hoja de Estilo en Cascada (CSS): es un lenguaje de estilo que define la presentación de los documentos HTML. Por ejemplo, abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas.CSS fue toda una revolución en el mundo del diseño web. [24]

Capítulo 1. Fundamentación teórica

JavaScript

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.[25]

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. [25]

Lenguajes utilizados del lado del servidor

PHP 5: es el acrónimo de Hypertext Preprocessor, es un lenguaje de código abierto interpretado de alto nivel, especialmente pensado para desarrollos web, el cual puede ser embebido en páginas HTML (HyperText Markup Language). Es ejecutado en el servidor. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil. Es soportado por la mayoría de los servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro server, y muchos otros. También cuenta con una extensión DBX (Data Base X, cualquier base de datos) de abstracción de base de datos que permite usar de forma transparente cualquier base de datos soportada por la extensión. (26)

Framework utilizado del lado del servidor

Symfony 2.7.17: es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. [26]

Symfony está desarrollado completamente en PHP. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de

Capítulo 1. Fundamentación teórica

gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Sistema gestor de Base de Datos

PostgreSQL 9.1: es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. [27]

Servidor Web

Nginx es un servidor web y proxy inverso de código abierto ligero de alto rendimiento, que también incluye servicios de correo electrónico con acceso al Internet Message Protocol (IMAP) y al servidor Post Office Protocol (POP). Además, NGINX está listo para ser utilizado como un proxy inverso. En este modo, NGINX se utiliza para equilibrar la carga entre los servidores back-end, o para proporcionar almacenamiento en caché para un servidor back-end lento [28].

Son muchas las características que ofrece este servidor web, pero una de las más importantes es que se trata de un software que es asíncrono, a diferencia de Apache que está basado en procesos. La ventaja principal de ser asíncrono, es su escalabilidad. En un sistema basado en procesos, cada conexión simultánea requiere de un hilo, lo que puede llevar a sobrecargar el servidor, mientras que en un servidor asíncrono se gestionan las peticiones en muy pocos hilos, reduciendo las posibilidades de sobrecarga en el servidor [28].

Entorno de Desarrollo Integrado

NetBeans IDE 7.1.3: es un Entorno de Desarrollo Integrado libre y gratuito sin restricciones de uso. Existe además un número importante de módulos para extenderlo.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica

Capítulo 1. Fundamentación teórica

como módulo. NetBeans permite crear aplicaciones web con PHP, un potente debugger integrado y además viene con soporte para Symfony un gran framework MVC escrito en php.[29]

1.7 Conclusiones parciales

Con la realización del presente capítulo han quedado definidos los conceptos más importantes para el correcto entendimiento de los términos usados en la presente investigación. Se realizó un estudio de los principales protocolos de intercambio y recuperación de información entre sistemas de información documental, donde se llegó a la conclusión de que el protocolo OAI-PMH, es el más idóneo para darle solución al problema propuesto en el presente trabajo de diploma debido a su simplicidad a la hora de implementar, cantidad de instituciones de archivo que lo han desarrollado entre otras. Se describieron además las herramientas, lenguajes y metodología a utilizar en el desarrollo del módulo.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

En el presente capítulo se presenta el modelo de dominio para ayudar a la comprensión del contexto del sistema, capturando los componentes más importantes con los que se trabajará. Se muestra los requerimientos funcionales a partir de los cuales se conforman las historias de usuario. Además, se presentan los diagramas relacionados con el análisis y el diseño de la solución que se propone.

2.1 Descripción de la propuesta de solución

La propuesta de solución consiste en incorporarle a la PREEPFV un módulo que permita:

- Recuperar metadatos disponibles en otros repositorios digitales bajo las especificaciones del protocolo para el intercambio y recuperación de información OAI-PMH.
- Almacenar los metadatos recuperados en la base de datos de la PREEPFV.
- Realizar búsquedas sencillas sobre los metadatos almacenados y poder visualizarlos.

2.2 Modelo de dominio

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en que trabaja el sistema. El modelo del dominio se describe mediante diagramas de UML (especialmente mediante diagramas de clases) [30].

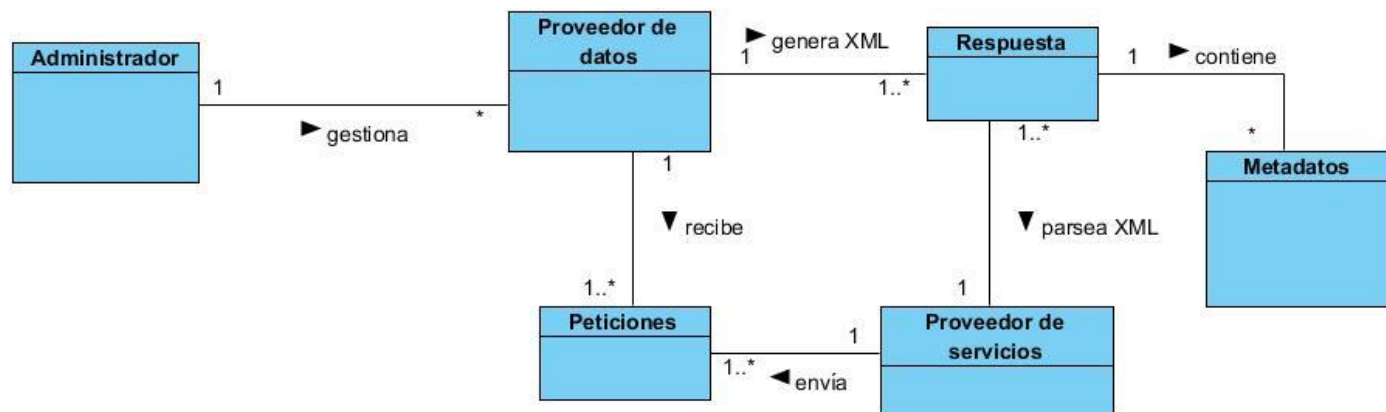


Fig. 1: Diagrama del modelo de dominio

Capítulo 2. Propuesta de solución

Conceptos que conforman el modelo de dominio:

Administrador: profesional responsable de agregar, modificar, editar y eliminar los repositorios digitales para la recolección de metadatos.

Proveedor de datos: maneja el depósito y la publicación de los recursos en un repositorio y expone los metadatos de los recursos del repositorio para que puedan ser recolectados.

Proveedor de servicios: realiza una petición al proveedor de datos para que le envíe metadatos según determinados criterios.

Respuesta: efecto, resultado, rendimiento, que se desea, se pretende o se espera de una cosa o de una persona.

Metadatos: son datos que describen otros datos.

Peticiones: es un requerimiento o solicitud que le hace un cliente a un servidor.

2.3 Especificación de requisitos de software

Requerimientos funcionales

Los requisitos funcionales describen como debe comportarse el sistema ante un determinado estímulo. Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones específicas [31]. A continuación, se presentan los requerimientos que debe cumplir el sistema:

Tabla 1: Requisitos Funcionales

No.	Requisito Funcional	Descripción
		Permite adicionar un nuevo repositorio OAI-PMH, insertando los siguientes datos: ➤ Nombre

Capítulo 2. Propuesta de solución

RF1	Adicionar repositorios OAI-PMH	<ul style="list-style-type: none">➤ Descripción➤ URL➤ Cantidad máxima a recolectar➤ Frecuencia de recolección
RF2	Modificar repositorios OAI-PMH	<p>Permite modificar los datos del repositorio OAI-PMH seleccionado. Los datos a modificar son los siguientes:</p> <ul style="list-style-type: none">➤ Nombre➤ Descripción➤ URL➤ Cantidad máxima a recolectar➤ Frecuencia de recolección
RF3	Eliminar repositorios OAI-PMH	Permite eliminar uno o varios repositorios.
RF4	Listar repositorios OAI-PMH	<p>Permite listar los repositorios que han sido adicionados. Cada repositorio muestra los siguientes datos:</p> <ul style="list-style-type: none">➤ Nombre➤ Descripción➤ URL

Capítulo 2. Propuesta de solución

		<ul style="list-style-type: none">➤ Cantidad máxima a recolectar➤ Frecuencia de recolección
RF5	Buscar documentos indexados	Permite realizar búsquedas sencillas sobre los documentos indexados de los repositorios. El criterio de búsqueda va a ser el título del metadato.
RF6	Visualizar metadatos.	El sistema visualiza los detalles del metadato seleccionado.
RF 7	Recolectar metadatos	Al ejecutar un comando la plataforma se conecta a los repositorios digitales agregados y recolecta los metadatos.

Requerimientos No Funcionales

Especifican criterios para evaluar la operación de un servicio de tecnología de información, en contraste con los requerimientos funcionales que especifican los comportamientos específicos[32].

Confiabilidad.

RNF 1. Sólo los usuarios con una combinación de nombre de usuario y contraseña válidos podrán acceder a la información de la plataforma. Restringir el acceso sólo para los usuarios que estén registrados en la plataforma.

Rendimiento.

RNF 2. Las consultas realizadas a la base de datos deben tener un tiempo de respuesta inferior a dos segundos para todos los usuarios que acceden.

Capítulo 2. Propuesta de solución

2.4 Historias de usuarios

Las historias de usuario (HU) es una de las técnicas que utiliza la metodología AUP en su versión UCI para especificar los requisitos del *software*, las cuales describen brevemente las características que el sistema debe poseer, utilizando para ello un lenguaje no técnico [33].

En el cuarto escenario de AUP, en su variante UCI los requisitos se administran en historias de usuarios por lo que para la solución propuesta se describieron un total de 7 Historias de usuario, de las cuales se presenta la correspondiente al RF1: Adicionar repositorios OAI-PMH, las demás se encuentran reflejadas en los anexos.

Tabla 2: HU del requisito Adicionar repositorios OAI-PMH

Número: 1	Nombre del requisito: Adicionar repositorios OAI-PMH
Programador: Alex Acosta Montejo.	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción: 1- Objetivo: Permitir incluir nuevos repositorios OAI-PMH en el sistema.	

Capítulo 2. Propuesta de solución

2- Acciones para lograr el objetivo (precondiciones y datos):

Para incluir un repositorio OAI-PMH es necesario:

- Estar autenticado en el sistema con el rol administrador.

3- Comportamientos válidos y no válidos (flujo central y alternos):

Los siguientes campos son obligatorios:

- Nombre: nombre del repositorio
- Descripción: breve descripción del repositorio.
- URL: dirección del repositorio OAI-PMH.
- Cantidad máxima a recolectar: cantidad máxima de datos que se deseen recolectar.
- Frecuencia con la cuál se recogerán los datos. Esta puede ser: diario, semanal o mensual.

4- Flujo de la acción a realizar:

- El sistema debe permitir al administrador incluir un repositorio OAI-PMH seleccionando la opción Incluir. Una vez seleccionada la opción debe especificar los siguientes datos: nombre, descripción, URL, cantidad máxima a recolectar y frecuencia de recolección.
- Cuando el usuario incluye correctamente los datos necesarios para incluir un repositorio OAI-PMH y selecciona la opción Guardar, se crea un nuevo elemento.
- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción en cuestión.
- Si selecciona la opción Cancelar regresará a la vista previa.

Observaciones:

Capítulo 2. Propuesta de solución

Prototipo de interfaz:

Nombre

Descripción

URL

Cantidad Máxima a recolectar

Frecuencia de recolección

Activado

2.5 Patrón Arquitectónico

Un patrón de arquitectura es una plantilla para una arquitectura de aplicaciones, especifica las propiedades generales a la estructura del sistema y repercute en la arquitectura de sus subsistemas. La selección de un patrón de arquitectura es por lo tanto una decisión fundamental al desarrollar un sistema de *software* [34].

Modelo-Vista-Controlador MVC es un patrón de arquitectura de *software* encargado de separar la lógica de negocio de la interfaz del usuario, separa los datos de una aplicación, la interfaz de usuario y la lógica de

Capítulo 2. Propuesta de solución

control en tres componentes distintos, este patrón facilita la funcionalidad, el mantenimiento y escalabilidad del sistema de forma simple y sencilla [35].

MVC divide las aplicaciones en tres niveles de abstracción [33]:

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

2.6 Patrones de diseño

Un patrón de diseño define un esquema de refinamiento de los subsistemas o componentes dentro de un sistema, o las relaciones entre estos. Este describe una estructura común y recurrente de componentes interrelacionados, que resuelve un problema general de diseño dentro de un contexto particular[34].

Para la implementación del módulo se utilizaron los patrones de diseño GRASP, acrónimo de *General Responsibility Assignment Software Patterns* (patrones generales de *software* para asignar responsabilidades). A continuación, se describen el uso de estos en el módulo desarrollado.

- **Experto:** La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. En la solución propuesta este patrón se evidencia en la clase entidad `MetadataRecolected()` [36].
- **Controlador:** consiste en asignar la responsabilidad de controlar el flujo de eventos del sistema [36]. Este patrón se utiliza en las clases controladoras que son las que se encargan de obtener datos y enviarlos a las vistas.
- **Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión

Capítulo 2. Propuesta de solución

posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases [36].

➤ **Alta Cohesión:** Plantea que la información que almacena una clase debe ser coherente y debe estar (en la medida de lo posible) relacionada con la clase [36].

La propia implementación del *framework* utilizado contiene estos patrones nivelados puesto que permite el uso de los componentes de forma individual, evidenciando el bajo acoplamiento, así como la dependencia entre ellos o alta cohesión.

2.7 Modelo de diseño.

Representa todas las clases del diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos, constituyendo la entrada principal a las actividades de la fase de implementación [37].

A continuación, se describe de manera general el significado de los principales elementos presentes en los diagramas de clases del diseño para lograr una mejor comprensión de estos:

Paquete Vista: Contiene las clases que muestran la información al usuario.

Paquete Controlador: Contiene las clases que realizan el tratamiento de eventos.

Paquete Modelo: Contiene las entidades generadas en correspondencia con las tablas de la base de datos que almacena toda la información que maneja el módulo.

A continuación, se presenta el diagrama de clases de diseño correspondiente:

Capítulo 2. Propuesta de solución

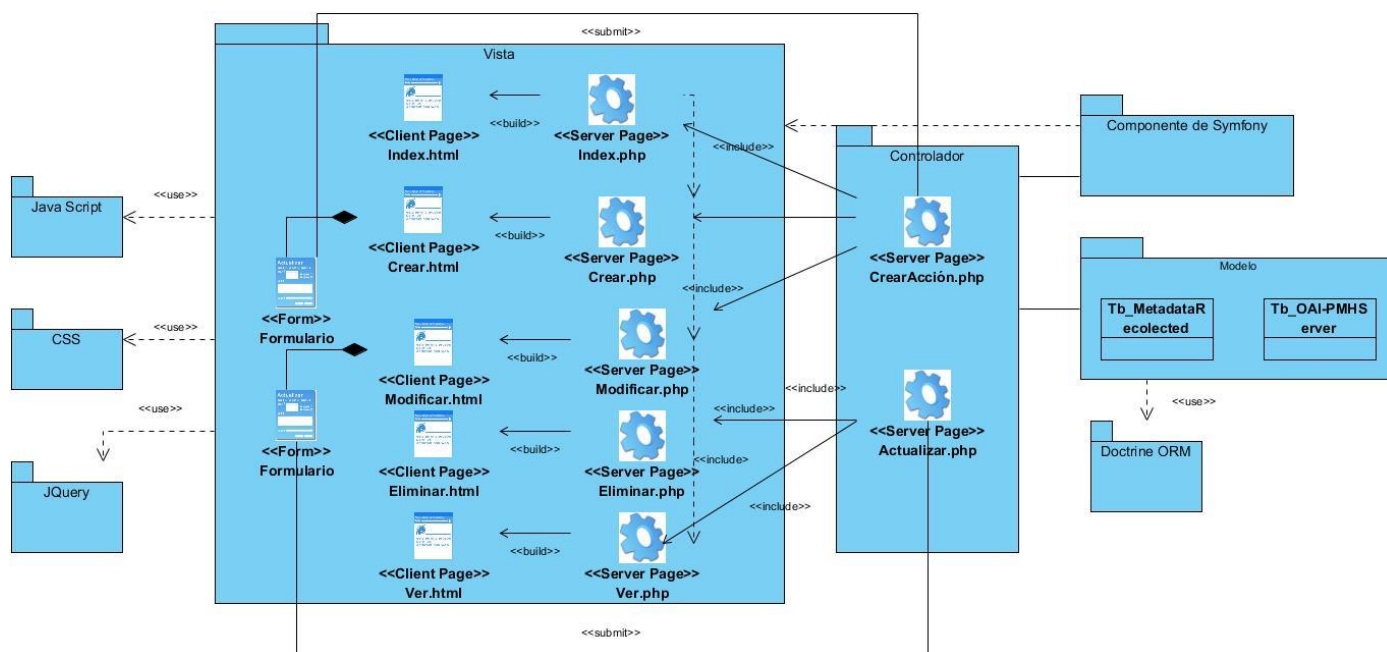


Fig. 2 Diagrama de clases .Elaboración propia.

2.8 Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados mediante enlaces de comunicación [38]. A continuación, se describen los elementos que componen el diagrama de despliegue correspondiente al presente trabajo de diploma.

- **Nodo PC Cliente:** se refiere a las computadoras que utilizarán los usuarios para interactuar con la aplicación. Se comunica con el Servidor de Aplicación a través del protocolo HTTP.
- **Nodo Servidor de Aplicación:** representa el servidor Apache donde se encuentra instalado el sistema.
- **Nodo Servidor de Base de Datos:** es el servidor Postgrespl donde se almacena la base de datos del sistema.
- **Nodo Base de Datos OAI-PMH:** es el servidor perteneciente al repositorio digital donde se recopilarán los metadatos.

Capítulo 2. Propuesta de solución

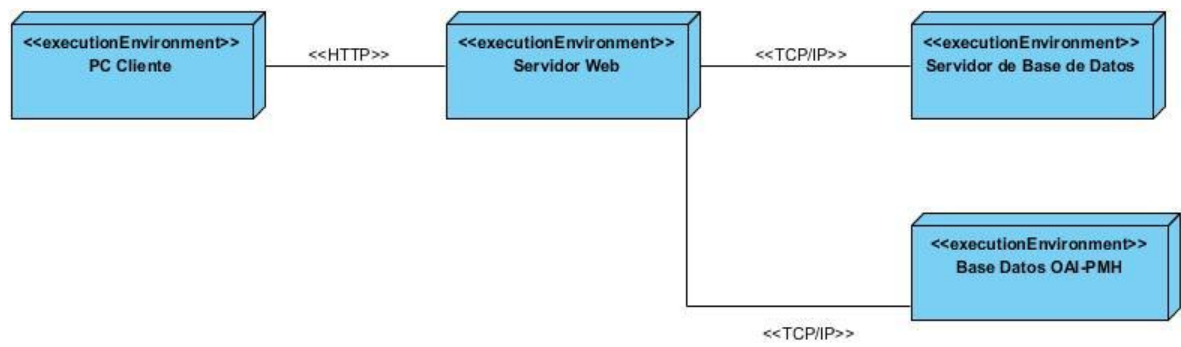


Fig. 3 Diagrama de despliegue. Elaboración propia.

2.9 Modelo de datos

Colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia[39].

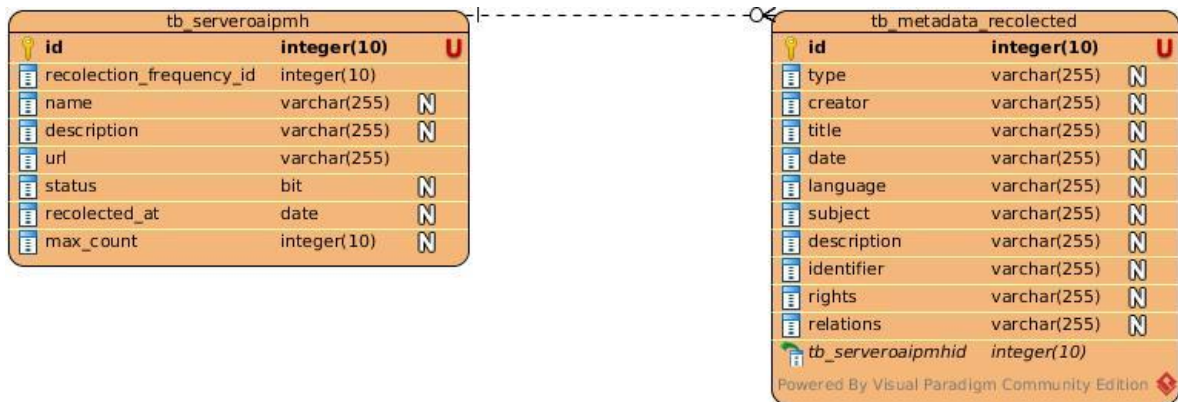


Fig. 4 Diagrama entidad-relación. Elaboración propia.

Descripción de las tablas de la base de datos

En esta sección se realiza la descripción de las tablas de la base de datos de la aplicación.

Tabla 3: Descripción de la tabla serveroaipmh

Tb_serveroaipmh
Descripción: Representa los atributos de servidor OAI-PMH

Capítulo 2. Propuesta de solución

Atributo	Tipo	Descripción
id	Intiger	Campo que contiene el identificador de la tabla.
name	Varchar	Campo que contiene el nombre del servidor.
description	Varchar	Campo que contiene la descripción del servidor.
url	Varchar	Campo que contiene la dirección url del servidor.
status	Bit	Campo que indica si el servidor se encuentra activo.
recolected_at	Date	Campo que indica la última vez que se recolecto información.
max_count	Intiger	Campo que indica la cantidad de datos a recolectar.

Tabla 4: Descripción de la tabla metadata-recolected

Tb_metadata_recolected
Descripción: Representa los atributos de los metadatos recolectados.

Capítulo 2. Propuesta de solución

Atributo	Tipo	Descripción
id	Intiger	campo que contiene el identificador de la tabla.
type	Varchar	Campo que contiene el tipo de metadato.
creator	Varchar	Campo que contiene nombre del creador del metadato.
title	Varchar	Campo que contiene el título del metadato.
date	Date	Campo que indica la fecha de recolección de datos.
language	Varchar	Campo que indica el lenguaje de los metadatos.
subject	Varchar	Campo que indica el autor del metadato.
description	Varchar	Campo que indica una descripción del metadato.
identifier	Varchar	Campo que indica el identificador del metadato.

Capítulo 2. Propuesta de solución

rights	Varchar	Campo que indica los derechos del metadato.
relation	Varchar	Campo que indica la relación del metadato.
tb_serveroaimh	Intiger	Campo que el id correspondiente al servidor en la tabla tb_serveroaimh.

2.10 Conclusiones parciales

- Con el estudio de los principales conceptos asociados al dominio, se modeló su representación para facilitar su comprensión.
- Se definió la arquitectura MVC como arquitectura a aplicar por las facilidades que proporciona, así como se utilizaron varios patrones para el diseño que facilitan la reutilización y mejor organización de la solución.
- Se modelaron artefactos ingenieriles, los cuales permitieron obtener una mayor comprensión del módulo, así como definir los principios que guiaron su implementación.

Capítulo 3. Implementación y Pruebas

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Una vez realizado el flujo de trabajo correspondiente al análisis y diseño, se tiene una visión más general de cómo se implementarán las funcionalidades del sistema, teniendo como precedente los artefactos generados y la descripción de cada uno de los requisitos. Además, se llevará a cabo el proceso de prueba a la solución desarrollada, un aspecto de vital importancia en el desarrollo del software propiciando así la obtención de buenos resultados, las cuales se ejecutan con el objetivo de revisar que el software tenga el nivel de calidad requerido y permiten detectar errores durante su funcionamiento. Este proceso debe comenzar en la fase de requerimientos y terminar con la finalización de la aplicación. En el proceso de pruebas se definen varios métodos, técnicas y tipos de pruebas, las cuales se abordarán durante el desarrollo del presente capítulo.

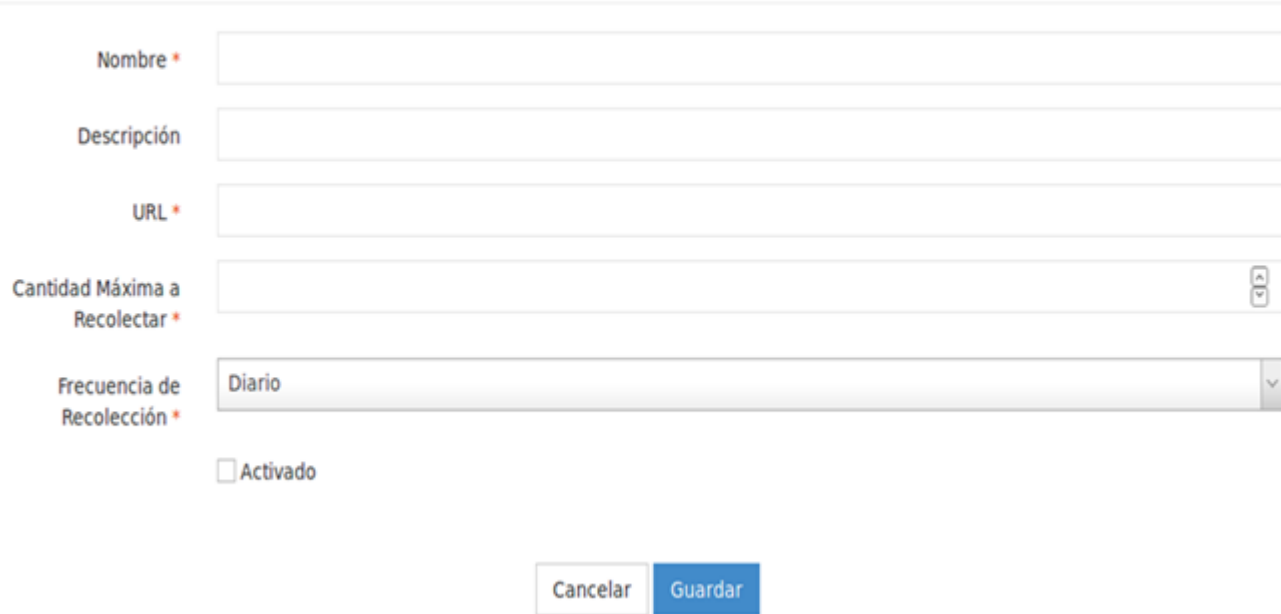
3.1 Modelo de implementación

El modelo de implementación garantizó la implementación de la propuesta de solución, basándose en su arquitectura, sus componentes, de acuerdo a su mecanismo de estructuración y los lenguajes de programación viables [40].

Descripción de la solución

Para poder realizar la recolección de metadatos es necesario poder dar de alta en el sistema a aquellos repositorios digitales de los cuales queremos obtener la información, para lograr esto se hace necesario llenar un formulario con los datos referentes a dicho repositorio que son: nombre del repositorio, breve descripción del repositorio, dirección del repositorio OAI-PMH, cantidad máxima de metadatos que se deseen recolectar, frecuencia con la cual se recogerán los datos, la cual puede ser: diario, semanal o mensual y guardar esto en la tabla de la base de datos llamada RepositorioOAIPMH. Ver fig 5.

Capítulo 3. Implementación y Pruebas



Nombre *

Descripción

URL *

Cantidad Máxima a Recolectar *

Frecuencia de Recolección *

Diario

Activado

Cancelar Guardar

Fig. 5 Formulario Crear Servidor OAI-PMH.

Para realizar la recolección de metadatos se programó una tarea que se encarga de ejecutar el método `HarvestMetadataRecolected()`, el cual recolecta los datos de aquellos repositorios incluidos anteriormente, guardando estos metadatos en la tabla `MetadataRecolected` en la base de datos.

Una vez recolectados los metadatos los usuarios pueden realizar búsquedas sencillas sobre la información disponible en la tabla `MetadataRecolected`. Ver fig.6.

Capítulo 3. Implementación y Pruebas

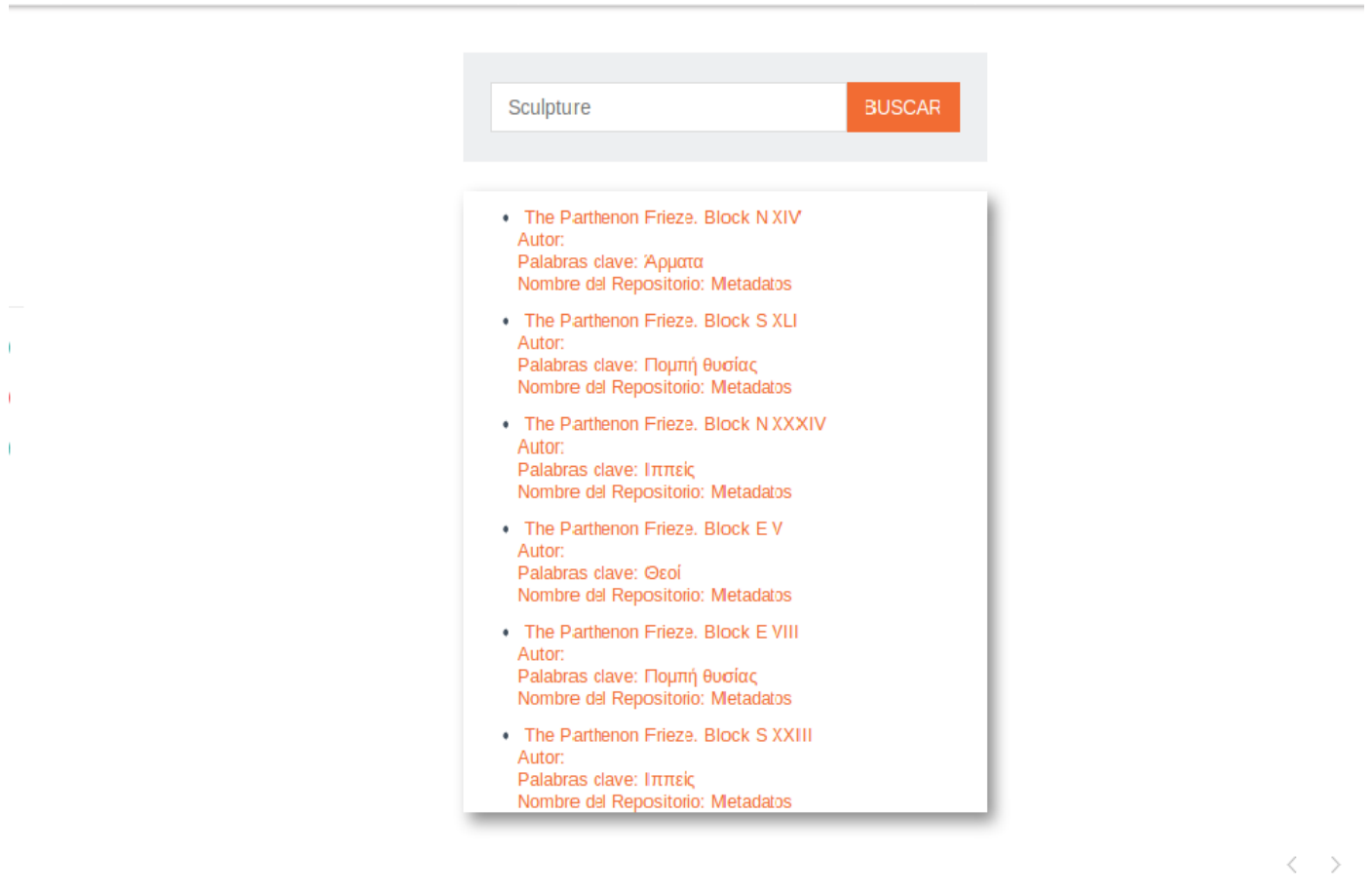


Fig. 6 Búsqueda de metadatos

Los usuarios pueden ver toda la información de un metadato seleccionándolo de la búsqueda realizada. Ver fig 7.

Capítulo 3. Implementación y Pruebas



Fig. 7 Detalles del metadato

Diagrama de componentes

El diagrama de componentes muestra la estructura de los componentes, incluyendo clasificadores que especifican componentes, y artefactos que los implementan. También se muestra la estructura de alto nivel del modelo de implementación en términos de subsistemas de implementación, y las relaciones entre elementos de implementación [41]. A continuación, se muestra el diagrama de componentes del sistema:

Capítulo 3. Implementación y Pruebas

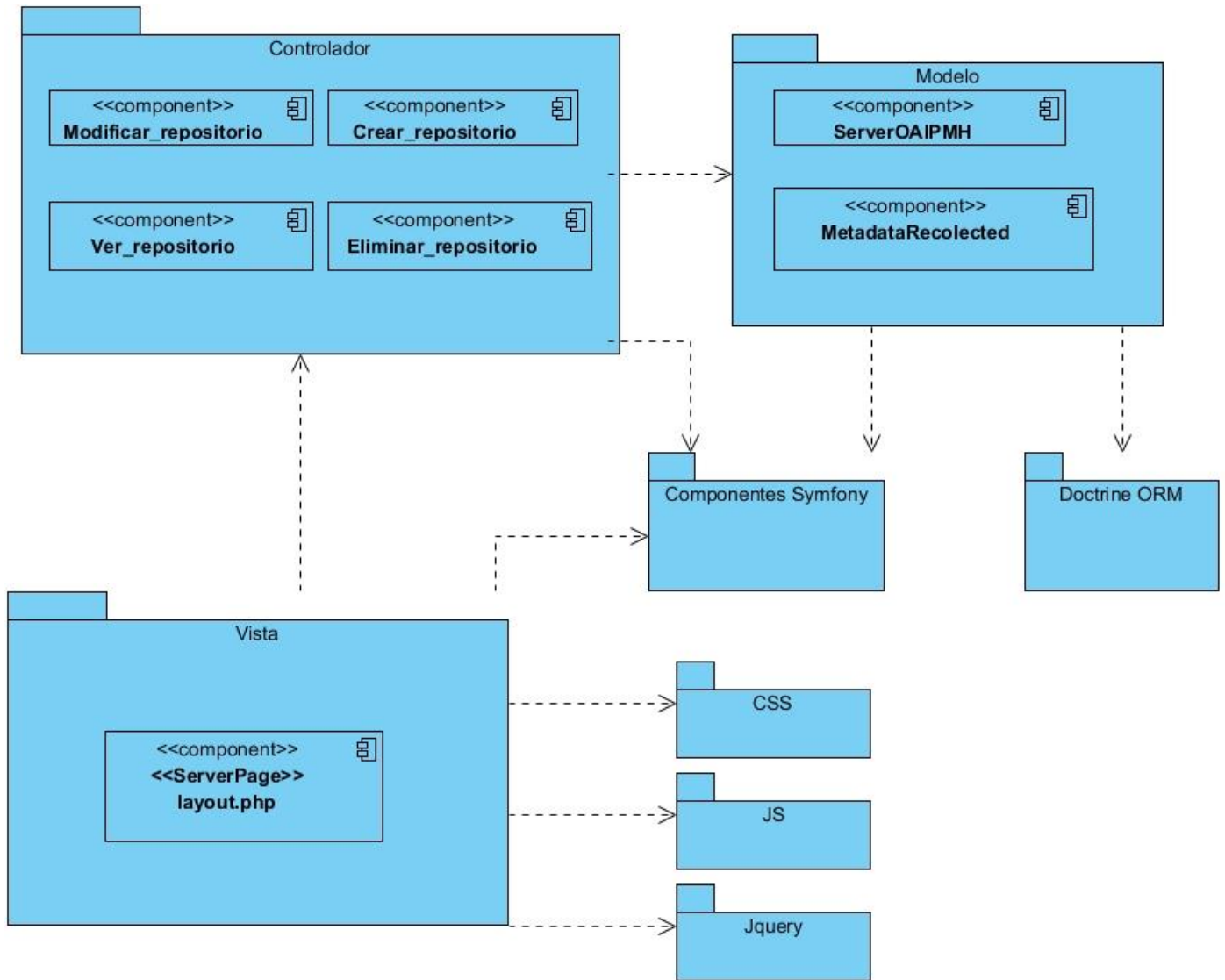


Fig. 8 Diagrama de Componentes. Elaboración propia

3.3 Pruebas de software

Por lo general se tiende a pensar que el proceso de pruebas consiste exclusivamente en ejecutar pruebas, es decir, ejecutar el software. Por supuesto, la ejecución de pruebas forma parte del proceso de pruebas, pero no presentan la totalidad de las actividades incluidas en él [41].

Las actividades de pruebas se dan antes y después de la ejecución de la prueba. Entre estas actividades se encuentran: planificar y controlar, seleccionar las condiciones de las pruebas, diseñar y ejecutar casos

Capítulo 3. Implementación y Pruebas

de pruebas, comprobar los resultados, evaluar los criterios de salida, elaborar informes sobre el proceso de pruebas y sobre el sistema probado, y finalizar o completar actividades de cierre una vez finalizada una fase de prueba [41].

El proceso de pruebas puede tener los siguientes objetivos [41]:

- ✓ Identificar defectos en el sistema.
- ✓ Aumentar la confianza y calidad del sistema.
- ✓ Facilitar información para la toma de decisiones.

3.4 Niveles de pruebas

Cuando se le van a aplicar pruebas a un software, se tienen en cuenta una serie de objetivos en diferentes escenarios y niveles de trabajo, debido a que las pruebas son agrupadas por niveles que se encuentran en distintas etapas del proceso de desarrollo [41]. Una vez implementado el sistema fue sometido al nivel de prueba que a continuación se detalla:

➤ Pruebas de sistema

Las pruebas de sistema deben estudiar los requisitos funcionales y no funcionales del sistema y las características de calidad de los datos. Los probadores también deben enfrentarse a requisitos incompletos o no documentados. Las pruebas de sistemas de los requisitos funcionales empiezan utilizando las técnicas basadas en la especificación (método de caja negra) más apropiadas para el aspecto del sistema a probar. Así, por ejemplo, puede crearse una tabla de decisión para las combinaciones de los efectos descritos en las normas de negocio. A continuación pueden utilizarse técnicas basadas en la estructura (método de caja blanca) para evaluar la exhaustividad de las pruebas por lo que representa a un elemento estructural, como por ejemplo una estructura de menú o la navegación de una página web [41].

Métodos de prueba

Un producto de ingeniería se puede probar mediante dos métodos de prueba [41]:

- Al conocer la función específica que se asignó a un producto para su realización, pueden llevarse a cabo pruebas que demuestren que cada función es completamente operativa mientras al mismo tiempo se buscan errores en cada función.

Capítulo 3. Implementación y Pruebas

- Conociendo el funcionamiento interno de un producto, pueden realizarse pruebas para garantizar que las operaciones internas se realizan de acuerdo con las especificaciones y que todos los componentes internos se revisaron de manera adecuada.

El primer enfoque de pruebas considera una visión externa y se llama prueba de caja negra. El segundo requiere una visión interna y se denomina prueba de caja blanca.

Para verificar el correcto funcionamiento será aplicado el método de caja negra, debido a que este permitirá corregir problemas en la interfaz del usuario y se comprobará que la propuesta de solución realiza las funciones requeridas por el usuario.

Método de caja negra:

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra realizan pruebas sobre la interfaz del programa a desarrollar, entiéndase por interfaz de entrada y salida del programa, no requieren del conocimiento de la lógica del programa y solamente se debe conocer la funcionalidad que este debe realizar [41].

Partición equivalente

La partición de equivalencia es una técnica de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Esta técnica permite examinar los valores válidos e inválidos de las entradas existentes en el software y descubrir de forma inmediata una clase de errores [41].

Diseño de caso de prueba

El diseño de caso de prueba es una parte de las pruebas de componentes y sistemas en las que se diseñan los casos de prueba (entradas y salidas esperadas) para probar el sistema. El objetivo del proceso de diseño de casos de prueba es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas para lograr que el programa satisfaga sus requerimientos [41].

Para diseñar un caso de prueba se selecciona una característica del sistema o componente que se desea

Capítulo 3. Implementación y Pruebas

probar. A continuación, se seleccionan un conjunto de entradas que ejecutan dichas características, documenta las salidas esperadas o rangos de salidas y donde sea posible se diseña una prueba automatizada que prueba que las salidas reales y esperadas son las mismas [41].

A continuación, se muestra el diseño de casos de prueba perteneciente al requisito funcional Adicionar repositorio OAI-PMH:

Capítulo 3. Implementación y Pruebas

Tabla 5: Caso de prueba perteneciente al RF Adicionar repositorio OAI-PMH

Tabla Caso de prueba: Adicionar repositorio OAI-PMH
CP Adicionar repositorio OAI-PMH
<p>Descripción general</p> <p>La funcionalidad inicia cuando el usuario decide adicionar un repositorio nuevo. Para ello el sistema muestra un formulario con los campos correspondientes al repositorio. Una vez llenado el formulario se selecciona la opción guardar y se guarda el repositorio en la base de datos.</p> <p>Condiciones de ejecución</p> <p>La persona debe estar autenticada en el sistema.</p>

Capítulo 3. Implementación y Pruebas

Escenario	Descripción	Nombre	URL	Descripción	Activado	Frecuencia de recolección	Cantidad máxima a recolectar	Respuesta del Sistema	Flujo Central
EC1.1	Selecciona en el calendario el día que desea crear el evento.	No aplica.	No aplica.	No aplica.	No aplica.	No aplica.	No aplica.	Se muestra un formulario con los siguientes campos: <ul style="list-style-type: none"> • Asunto • Contenido • Fecha de inicio • Fecha de fin • Repetir 	Calendario/Nuevo evento

Capítulo 3. Implementación y Pruebas

								<ul style="list-style-type: none"> Recordar 		
EC1.2	Introduce los datos y selecciona "Guardar".	V	V	V	V	V	V	<p>Valida los datos.</p> <p>Registra un nuevo evento en el sistema.</p>	Plataforma/ Incluir/ Aceptar	
EC1.3	Seleccione la opción "Cancelar"	No aplica.	No aplica.	No aplica.	No aplica.	No aplica.	No aplica.	<p>Elimina los datos.</p> <p>Regresa a la vista anterior.</p>	Plataforma/ Incluir/Cancelar	
EC1.4	Existen campos vacíos.	I	V	V	V	V	V	<p>Muestra un mensaje informando: "Verifique que</p>	Plataforma/ Incluir/ Aceptar	
		V	I	V	V	V	V			
		V	V	I	V	V	V			

Capítulo 3. Implementación y Pruebas

		V	V	V	I	V	V	no existan campos vacíos". Muestra un indicador sobre el campo vacío. Regresa al 1.2	
		V	V	V	V	I	V		
		V	V	V	V	V	I		

Capítulo 3. Implementación y Pruebas

3.5 Resultados obtenidos en las pruebas

Al módulo de recuperación de metadatos para la Plataforma de Recursos Empresa Editorial Poligráfica Félix Varela se le realizó la verificación del cumplimiento de los requisitos funcionales establecidos mediante uso de las Pruebas de Caja Negra, teniendo en cuenta la técnica de partición por equivalencia, la cual se basa en la definición de casos de pruebas, que tienen como objetivo detectar la mayor cantidad de no conformidades posibles en las funcionalidades del sistema.

A continuación, se presentan el resultado de las pruebas realizadas:

Tabla 6: Resultados de las pruebas del sistema

Iteraciones	Cantidad de casos de prueba	No conformidades detectadas			
		Alta	Media	Baja	Total
1	7	3	8	15	26
2	7	1	5	7	13
3	7	0	0	0	0

Tabla 7: Clasificación de las No Conformidades por el tipo de error

Clasificación de la NC	Tipo de NC	Descripción
Funcionalidad	Alta	El resultado Mostrado al realizar una acción no es el esperado.
Errores de Idioma	Alta/Media	Errores relativos al uso de dos o más idiomas en la aplicación.
Ortografía	Baja	Existen palabras con errores ortográficos, por ejemplo cambios de consonantes o vocales repetidas, abreviaturas mal

Capítulo 3. Implementación y Pruebas

		empleadas, palabras sin tildar, utilización incorrecta de mayúsculas y minúsculas.
--	--	--

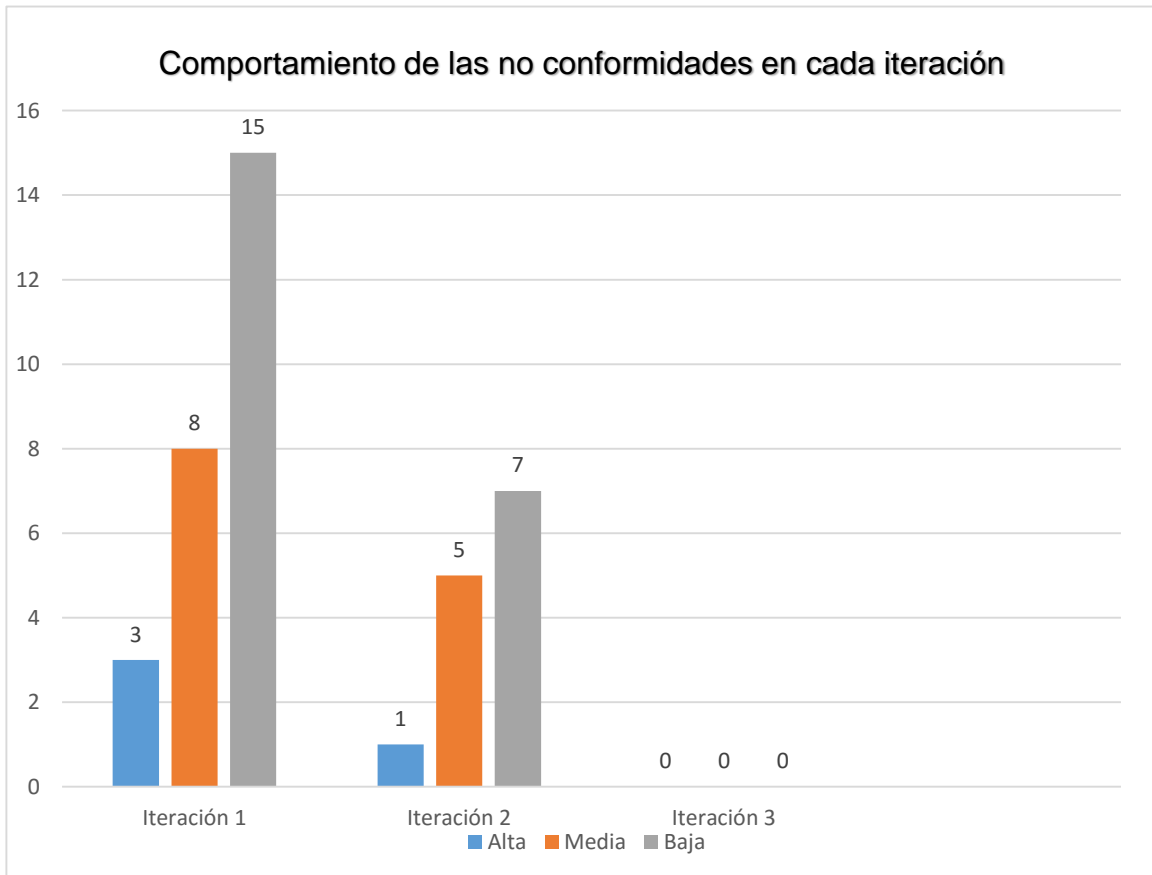


Fig. 9 No conformidades detectadas

➤ Pruebas de aceptación

Las pruebas de aceptación son importantes dado que significan la satisfacción del cliente con el producto desarrollado, el final de una iteración y el comienzo de la siguiente. Se elaboran a lo largo de la iteración, en paralelo con el desarrollo del sistema, y adaptándose a los cambios que el sistema sufra [41].

Capítulo 3. Implementación y Pruebas

3.6 Conclusiones parciales

Los artefactos generados durante la disciplina de implementación permitieron entender la relación de dependencia y por último el desarrollo final de la solución. Además, se aplicaron pruebas de sistema y de aceptación, permitiendo determinar y erradicar las deficiencias encontradas, obteniendo finalmente una solución con un alto nivel de calidad.

Conclusiones

Una vez finalizada la investigación se puede arribar a las siguientes conclusiones:

- La elaboración del marco teórico de la investigación mediante el estudio del estado del arte acerca de las tendencias tecnológicas actuales y las estrategias utilizadas para la recuperación de metadatos sustentó los principales referentes teóricos en los que se enmarca el desarrollo de la propuesta de solución.
- Se desarrolló un módulo para la Plataforma de recursos de la Empresa Editorial Poligráfica *Félix Varela* que permite la recuperación de metadatos desde repositorios digitales.
- Se realizaron pruebas al sistema mediante la técnica de caja negra, detectando un total de 39 no conformidades en tres iteraciones, las cuales fueron resueltas al finalizar cada iteración, mejorando la calidad del sistema.

Recomendaciones

Recomendaciones

- Implementar funcionalidades para que el módulo sea capaz de recuperar metadatos en otros formatos como marc, merts y rda.
- Implementar un proveedor de datos para la plataforma de recursos de Empresa Editorial Poligráfica Félix Varela.

Recomendaciones

Bibliografía

BIBLIOGRAFÍA

- [1] «Metadatos y gestión documental - neodoc». [En línea]. Disponible en: <http://www.neodoc.es/news/metadatos-y-gestion-documental>. [Accedido: 11-may-2018].
- [2] «Diccionario - dta - Ministerio de Educación, Cultura y Deporte». [En línea]. Disponible en: https://www.mecd.gob.es/cultura/areas/archivos/mc/dta/diccionario.html?searchfield_hidden=%23inicio&searchfield=archivo. [Accedido: 29-nov-2017].
- [3] M. M. Mujica, Gestión documental y organización de archivos. Editorial Félix Varela, 2005.
- [4] L. Carpenter, «1. OAI para principiantes: Introducción». [En línea]. Disponible en: <http://travesia.mcu.es/portaln/jspui/html/10421/1823/page1.htm#section2>. [Accedido: 11-dic-2017].
- [5] L. F. Gómez, «Interoperabilidad en los Sistemas de Información Documental (SID): la información debe fluir.», Códices Rev. Fac. Cienc. Económicas Soc. Programa Sist. Inf. Doc., vol. 3, n.º 1, pp. 23-39, 2007.
- [6] L. Carpenter, «6. Glosario». [En línea]. Disponible en: <http://travesia.mcu.es/portaln/jspui/html/10421/1823/page6.htm#section16>. [Accedido: 11-dic-2017].
- [7] L. Nelida, E. García, y A. Sergio Caballero, «Metadatos: necesidad e importancia de integrar estándares», dic. 2017.
- [8] «Dienst». [En línea]. Disponible en: http://www.sedic.es/autoformacion/acceso_abierto/4-definicion5.html. [Accedido: 11-dic-2017].
- [9] T. Krichel, «Guildford Protocol». [En línea]. Disponible en: http://openlib.org/acmes/root/docu/guilp_1999-12-27.html. [Accedido: 11-dic-2017].
- [10] «Simple Digital Library Interoperability Protocol». [En línea]. Disponible en: <http://diglib.stanford.edu:8091/~testbed/doc2/SDLIP/>. [Accedido: 11-dic-2017].
- [11] A. Guajardo Salinas, «Z39.50 y OAI-PMH: Protocolos de Transferencia y Rec uperación de Información», presentado en XV CONFERENCIA INTERNACIONAL DE BIBLIOTECOLOGIA "Panorama

Bibliografía

de las Bibliotecas y la Información en el bicentenario” 2, 3 y 4 de Noviembre de 2010. Centro Cultural Estación Mapocho.

[12] J. M. Barrueco y I. Subirats Coll, «Artículo Open archives initiative. Protocol for metadata harvesting (OAI-PMH): descripción, funciones y aplicaciones de un protocolo», El Prof. Inf. Vol 12 N° 2 Marzo-abril 2003.

[13] L. Carpenter, «2. Historia y desarrollo del OAI-PMH». [En línea]. Disponible en: <http://travesia.mcu.es/portaln/jspui/html/10421/1823/page2.htm>. [Accedido: 10-dic-2017].

[14] S. Ramírez y J. Pablo, «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 contenidos / Study, analysis, evaluation and implementation of a protocol for exchanging files over the internet for the interoperability of digital library repositories in the Universidad Nacional de Colombia with a content provider», Universidad Nacional de Colombia, 2010.

[15] M. J. L. Lapuente, «Metadatos Dublin Core». [En línea]. Disponible en: http://www.hipertexto.info/documentos/dublin_core.htm. [Accedido: 12-dic-2017].

[16] J. PEMAU ALONSO y J. A. BARROSO COTORRO, «INTRODUCCIÓN A OAI-PHM Y SU IMPLANTACIÓN EN EL PORTAL E-REVISTAS - PDF». [En línea]. Disponible en: <http://docplayer.es/14411762-Introduccion-a-oai-phm-y-su-implantacion-en-el-portal-e-revistas.html>. [Accedido: 12-dic-2017].

[17] «DSpace: un manual específico para gestores de la información y la documentación». [En línea]. Disponible en: <http://bid.ub.edu/20rodri2.htm>. [Accedido: 05-jun-2018].

[18] «Sierra, un gestor de bibliotecas open source de última generación - Grup de Treball de Programari Lliure per als Professionals de la Informació». .

[19] I. Sommerville, Ingeniería de Software. Madrid.

[20] T. R. Sánchez, «Metodología de desarrollo para la Actividad productiva de la UCI». .

[21] «About the Unified Modeling Language Specification Version 2.0». [En línea]. Disponible en: <http://www.omg.org/spec/UML/2.0/>. [Accedido: 23-ene-2018].

Bibliografía

- [22] «Visual Paradigm», calameo.com. [En línea]. Disponible en: <http://www.calameo.com/read/00038464516a006851891>. [Accedido: 23-ene-2018].
- [23] «HTML5», Documentación web de MDN. [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/HTML/HTML5>. [Accedido: 05-jun-2018].
- [24] «All CSS specifications». [En línea]. Disponible en: <https://www.w3.org/Style/CSS/specs>. [Accedido: 23-ene-2018].
- [25] «Programming languages used on the Internet and the World Wide Web (WWW)». [En línea]. Disponible en: <https://www.webdevelopersnotes.com/languages-on-the-internet>. [Accedido: 23-ene-2018].
- [26] F. Potencier y Z. Francois, «Symfony , la guía definitiva». [En línea]. Disponible en: http://librosweb.es/libro/symfony_1_4/. [Accedido: 23-ene-2018].
- [27] «PostgreSQL: PostgreSQL 9.1 released». [En línea]. Disponible en: <https://www.postgresql.org/about/news/1349/>. [Accedido: 23-ene-2018].
- [28] «NGINX, La Alternativa de un servidor web», Centripio, 06-oct-2017. .
- [29] «NetBeans IDE entorno de desarrollo para lenguajes como Java PHP C/C++ Groovy». [En línea]. Disponible en: <https://www.genbetadev.com/herramientas/netbeans-1>. [Accedido: 23-ene-2018].
- [30] «Modelo de Dominio | Tecnología y Synergix». [En línea]. Disponible en: <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>. [Accedido: 14-mar-2018].
- [31] P. por pmoinformatica.com, «Requerimientos funcionales: Ejemplos». .
- [32] «Requerimientos No Funcionales: Porque son importantes - La Oficina de Proyectos de Informática». [En línea]. Disponible en: <http://www.pmoinformatica.com/2013/01/requerimientos-no-funcionales-porque.html>. [Accedido: 14-mar-2018].
- [33] «Historias de Usuario», José Manuel Beas, 23-may-2011. .
- [34] «PPArquitecturaNET.pdf». .
- [35] «MVC_con_PHP.pdf». .
- [36] Fernando Alfonso Casas De la Torre, «Patrones GRASP», 04:58:54 UTC.

Bibliografía

- [37] «UNIDAD 4 MODELO DE DISEÑO». [En línea]. Disponible en: <http://oscarhdeztorresunidad4.blogspot.com/>. [Accedido: 05-jun-2018].
- [38] Q. Limachi Nancy Susana y (primero)Marca Huallpara Marca Huallpara Hugo Michael, «Diagrama de Despliegue.» .
- [39] «Qué es un modelo de base de datos | Lucidchart». [En línea]. Disponible en: <https://www.lucidchart.com/pages/es/qu%C3%A9-es-un-modelo-de-base-de-datos>. [Accedido: 14-mar-2018].
- [40] P. por L. hern y ez, «MODELO DE IMPLEMENTACIÓN» . .
- [41] R. S. Pressman, Ingeniería del software: un enfoque práctico, Séptima edición. .

ANEXOS

Tabla 8: HU Modificar repositorios OAI-PMH

<p>Número: 2</p>	<p>Nombre del requisito: Modificar repositorios OAI-PMH</p>	
<p>Programador: Alex Acosta Montejo.</p>		<p>Iteración Asignada: 1era</p>
<p>Prioridad: Alta</p>		<p>Tiempo Estimado: 2 días</p>
<p>Riesgo en Desarrollo: N/A</p>		<p>Tiempo Real:</p>
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir editar los repositorios OAI-PMH en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para editar un repositorio OAI-PMH hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <p>Los siguientes campos son obligatorios:</p>		

-Nombre: nombre del repositorio

-Descripción: breve descripción del repositorio.

-URL: dirección del repositorio OAI-PMH.

-Cantidad Máxima a recolectar: cantidad máxima de datos que se deseen recolectar.

-Frecuencia de recolección: (diario, semanal o mensual).

4- Flujo de la acción a realizar:

- Cuando el usuario selecciona la opción "Editar", el sistema debe permitir editar los datos de un repositorio OAI-PMH.

- Cuando el usuario incluye correctamente los datos y selecciona la opción "Actualizar", se actualiza el elemento.

- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción en cuestión.

- Si selecciona la opción Cancelar regresará a la vista previa.

Observaciones:

Prototipo de interfaz:

The image shows a web form with the following elements:

- Nombre:** A text input field.
- Descripción:** A text input field.
- URL:** A text input field.
- Cantidad Máxima a recolectar:** A small text input field.
- Frecuencia de recolección:** A dropdown menu with a downward arrow.
- Activado:** A checkbox.
- Cancelar:** A button.
- Actualizar:** A button.

Tabla 9: HU Eliminar repositorios OAI-PMH

Número: 3	Nombre del requisito: Eliminar repositorios OAI-PMH.
Programador: Alex Acosta Montejo.	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 4 días

<p>Riesgo en Desarrollo: N/A</p>	<p>Tiempo Real:</p>
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir eliminar un repositorio OAI-PMH en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para eliminar una notificación hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un repositorio. <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <p>N/A</p> <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El sistema permite eliminar una o varios repositorios, para eliminar varios repositorios el sistema muestra un listado de los mismos permitiendo marcar más de un elemento a eliminar, selecciona la opción eliminar de las acciones en lote y presiona la opción “Aplicar”, luego el sistema muestra un mensaje de confirmación y al seleccionar la opción “Aceptar” se eliminará el repositorio, si selecciona la opción “Cancelar”, se revocará el proceso. 	

	Nombre	URL	Activado	Acciones
<input type="checkbox"/>	Metadatos	http://repository.parthenonfrieze.gr/frieze-oai/request	SI	Ver, Editar, Eliminar

Tabla 10: HU Listar repositorios OAI-PMH

Número: 4	Nombre del requisito: Listar repositorios OAI-PMH
Programador: Alex Acosta Montejo.	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 6 días
Riesgo en Desarrollo: N/A	Tiempo Real:
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir listar los repositorios OAI-PMH en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para listar los repositorios hay que:</p>	

- Estar autenticado en el sistema con el rol X.

3- Comportamientos válidos y no válidos (flujo central y alternos):

N/A

4- Flujo de la acción a realizar:

Quando el usuario va agregando nuevos repositorios, estos se van listando, dándole la posibilidad de elegir al usuario los repositorios que desee. Además, el usuario tiene la posibilidad de ver detalles de un repositorio, editarlo, y eliminarlo.


Observaciones:

Prototipo:

	Nombre	URL	Activado	Acciones
<input type="checkbox"/>	Metadatos	http://repository.parthenonfrieze.gr/frieze-oai/request	SI	Ver, Editar, Eliminar

Tabla 11: HU Buscar documentos indexados

Número: 5	Nombre del requisito: Buscar documentos indexados.
Programador: Alex Acosta Montejo.	Iteración Asignada: 1era

Prioridad: Media	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción: 1- Objetivo: Permitir buscar los datos de los documentos indexados en el sistema 2- Acciones para lograr el objetivo (precondiciones y datos): Para buscar en un documento hay que: - Tener en cuenta los siguientes datos: nombre y descripción. - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un documento. 3- Comportamientos válidos y no válidos (flujo central y alternos): El campo búsqueda es obligatorio: puede ser (título del documento). 4- Flujo de la acción a realizar: Cuando el usuario inserta un criterio de búsqueda (título del documento), el sistema muestra un listado de los documentos que cumplen con los criterios establecidos, en caso de no encontrar ningún elemento el sistema muestra un mensaje de información.	
	

Número: 6	Nombre del requisito: Mostrar los resultados de la búsqueda
Programador: Alex Acosta Montejo	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real:
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir consultar los resultados de la búsqueda en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para mostrar los resultados hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos documento encontrado. <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <p>N/A</p> <p>4- Flujo de la acción a realizar:</p> <p>Se muestra al usuario los documentos indexados después de realizar la búsqueda.</p>	

Título: North frieze
 Autor:
 Fecha:
 Lenguaje:
 Tema: frieze
 Identificador: oai:repository.pathenorfrieze.gr/frieze
 Tipo:
 Relaciones: <http://repository.pathenorfrieze.gr/frieze>

Tabla 12: HU Recolectar metadatos

Número: 7	Nombre del requisito: Recolectar metadatos.
Programador: Alex Acosta Montejo	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 2 días

Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción:	
1- Objetivo:	
Permitir recolectar metadatos.	
2- Acciones para lograr el objetivo (precondiciones y datos):	
N/A	
3- Comportamientos válidos y no válidos (flujo central y alternos):	
N/A	
4- Flujo de la acción a realizar:	
Al ejecutar un comando la plataforma se conecta a los repositorios digitales agregados y recolecta los metadatos	

Tabla 13: Caso de prueba perteneciente al RF Buscar documentos indexados

Tabla 8 Caso de prueba Buscar documentos indexados.
CP Buscar documentos indexados.
Descripción general Cuando el usuario inserta un criterio de búsqueda (título del documento), el sistema muestra un listado de los documentos que cumplen con los criterios establecidos, en caso de no encontrar ningún elemento el sistema muestra un mensaje de información.
Condiciones de ejecución La persona debe estar autenticada.

Escenario	Descripción	Título del documento	Respuesta del Sistema	Flujo Central
EC4.1 Buscar documentos indexados	Selecciona en la Plataforma la opción “Búsqueda de metadatos recolectados”		Se muestra una vista para insertar el criterio de búsqueda correspondiente: <ul style="list-style-type: none"> • Título del document 	Plataforma/BuscarDocumentosIndexados
EC4.2	Se introduce el criterio de búsqueda correspondiente al evento	V	El sistema muestra una ventana con los datos correspondientes: <ul style="list-style-type: none"> • Título • Autor • Fecha • Lenguaje 	Plataforma/BuscarDocumentosIndexados/Buscar

			<ul style="list-style-type: none"> • Tema • Identificador • Tipo • Relaciones 	
EC4.3	El usuario introduce un criterio de búsqueda incorrecto.		El sistema no muestra ningún documento indexado.	Plataforma/BuscarDocumentosIndexados/Buscar

Tabla 14: Caso de prueba perteneciente al RF Mostrar los resultados de la búsqueda

Tabla 8 Caso de prueba Mostrar los resultados de la búsqueda

CP Mostrar los resultados de la búsqueda

Descripción general

Cuando el usuario realiza una búsqueda se muestra al usuario los documentos indexados después de realizar la misma.

Condiciones de ejecución

La persona debe estar autenticada.

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC5.1 Mostrar los resultados de la búsqueda.	Al insertar un criterio de búsqueda en la opción "Búsqueda de metadatos recolectados" .	El sistema muestra una ventana con los datos correspondientes: <ul style="list-style-type: none"> • Título • Autor • Fecha 	Plataforma/BuscarDocumentosIndexados/MostrarResultadosde laBúsqueda

		<ul style="list-style-type: none"> • Lenguaje • Tema • Identificador • Tipo • Relaciones 		
--	--	---	--	--

Tabla 15: caso de prueba perteneciente al RF Listar repositorios OAI-PMH

Tabla 8 Caso de prueba Listar repositorios OAI-PMH
CP Listar repositorios OAI-PMH

Descripción general

Cuando el usuario va agregando nuevos repositorios, estos se van listando, dándole la posibilidad de elegir al usuario los repositorios que desee. Además, el usuario tiene la posibilidad de ver detalles de un repositorio, editarlo, y eliminarlo.

Condiciones de ejecución

La persona debe estar autenticada.

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC5.1 Listar repositorios OAI-PMH	El usuario inserta un nuevo repositorio OAI-PMH en la opción "incluir".	El sistema muestra un listado de los repositorios incluidos por el usuario, dándole, además, la posibilidad de ver, eliminar y editar el repositorio.	Plataforma/ Listar repositorios OAI-PMH

Tabla 16: Caso de prueba perteneciente al RF Eliminar repositorio OAI-PMH

Tabla Caso de prueba Eliminar repositorio OAI-PMH

CP Eliminar repositorio OAI-PMH

Descripción general

La funcionalidad inicia cuando el usuario decide eliminar un repositorio OAI-PMH de la lista de repositorios disponibles. Para ello el sistema muestra un listado con todos los repositorios en cuestión, si selecciona la opción eliminar de las acciones en lote y presiona la opción “Aplicar”, se mostrará un mensaje de confirmación y al seleccionar la opción “Aceptar” se eliminará el repositorio, si selecciona la opción “Cancelar”, se revocará el proceso.

Condiciones de ejecución

La persona debe estar autenticada.

Escenario	Descripción	Nombre	URL	Descripción	Activado	Frecuencia de recolección	Cantidd máxima a recolectar	Respuesta del Sistema	Flujo Central
EC2.1	Selecciona en la plataforma la opción							Se muestra una lista con todos los repositorios disponibles.	Plataforma/Listar repositorios

Anexos

	repositorios OAI-PMH.								
EC2.2	Se selecciona el repositorio q desee eliminar							Se muestra una lista con todos los repositorios y la opción para eliminarlos.	Plataforma/Eliminar repositorio
EC2.3	Selecciona la opción "Eliminar".							Elimina el repositorio seleccionado.	Plataform/Eliminar

Tabla 17: Caso de prueba perteneciente al RF Modificar repositorio OAI-PMH

Tabla Caso de prueba modificar repositorio OAI-PMH
CP Modificar repositorios OAI-PMH
<p>Descripción general</p> <p>La funcionalidad inicia cuando el usuario decide editar un repositorio creado. Para ello el sistema muestra un formulario con los campos correspondientes a la solicitud. Una vez llenado el formulario se selecciona la opción aceptar y se guardan los cambios en la base de datos.</p>
<p>Condiciones de ejecución</p> <p>La persona debe estar autenticada.</p>

Escenario	Descripción	Nombre	URL	Descripción	Activado	Frecuencia de recolección	Cantidad máxima a recolectar	Respuesta del Sistema	Flujo Central
EC3.1	Selecciona en el listado de repositorios el repositorio a editar.							<p>Se muestra un formulario con los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre • URL • Descripción • Activado • Frecuencia de 	Plataforma/Editar repositorio

									recolección <ul style="list-style-type: none"> • Cantidad máxima a recolectar 	
EC3.2	Introduce los datos y selecciona "Aceptar".	V	V	V	V	V	V	V	Valida los datos. Registra los datos nuevos del evento en el sistema.	Plataforma/Editar/Aceptar
EC3.3	Seleccione la opción "Cancelar"								Elimina los datos.	Plataforma/Editar

								Regresa a la vista anterior.	/Cancelar
EC3.4	Existen campos vacíos.	I	V	V	V	V	V	Muestra un mensaje informando: "Verifique que no existan campos vacíos". Muestra un indicador sobre el campo vacío. Regresa al 1.2	Plataforma/Editar /Aceptar
		V	I	V	V	V	V		
		V	V	I	V	V	V		
		V	V	V	I	V	V		
		V	V	V	V	I	V		
		V	V	V	V	V	I		

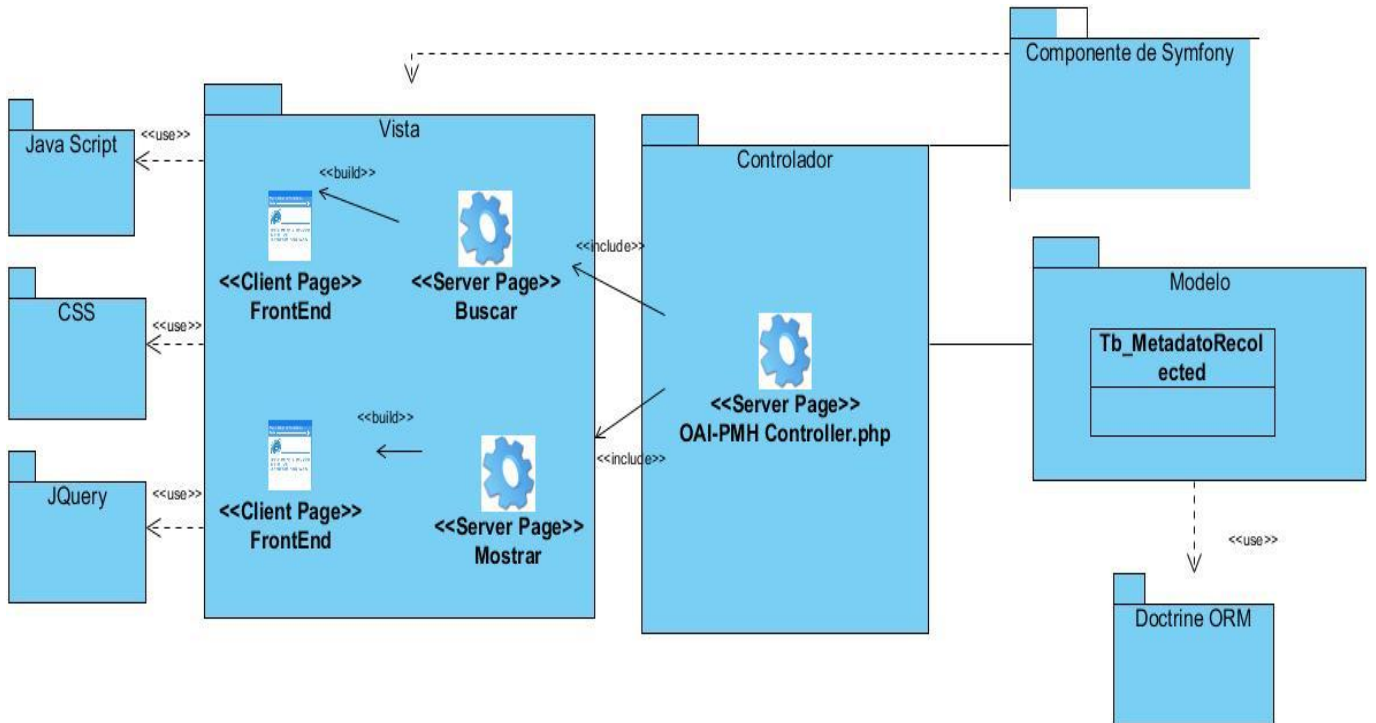


Fig. 10 Diagrama de clases Mostrar y Buscar metadatos