Implementación Del Repositorio De Datos DSpace De Modo Descentralizado E Incluyendo Protocolos De Descubrimiento Y Trasmisión De Datos

Alexander Martínez Méndez

Trabajo de grado para optar al título de
Ingeniero de Sistemas

Director: **Ph.D. Luis A. Núñez**

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingeniería de Sistemas e Informática
Bucaramanga

2017

Agradecimientos

Mucha personas han sido fundamentales en este proceso de formación, sin la contribución de cada una de ellas el recorrido hubiese sido en una cicla de acero y no de carbono. Agradecimiento especial a mi familia, por su apoyo incondicional, Padre, Madre y Hermanas, infinitas gracias; al profesor Luis, por su confianza y por plantar en mí el gusto por la ciencia; a CUSOL, por ayudarme a conocer el mundo del Software Libre; al grupo Halley, por permitirme viajar en la nave del conocimiento del Cosmos; a Eliecer, por la compañía en tantas aventuras y desaventuras; finalmente, a la universidad, por cada uno de los espacios de formación dentro y fuera de las aulas.

Contenido

Introdución **13**

1 Definición Del Problema **14**

2 Estado Del Arte **16**
2.1 Repositorios Digitales 16

3 Objetivos **18**
3.1 Objetivo General 18
3.2 Objetivos Específicos 18

4 Marco Teórico **19**
4.1 Repositorio Institucional 19
4.2 DSpace 19
4.3 Protocolos De Transmisión Y Descubrimiento De Datos 24
4.3.1 TCP/IP. 24
4.3.2 OAI-PMH. 24
4.3.3 SWORD. 25
4.4 PID (Persistent Identifiers) 26
4.5 WCD (Water Cherenkov Detectors) 26
4.6 MetaDatos 27
4.6.1 Dublin Core. 27
4.7 Virtualización 27
4.7.1 Hipervisores. 27
4.8 Demonio Cron 28
4.8.1 Uso Del Demonio Cron. 28

5 Metodología **29**

6 Desarrollo Del Proyecto **31**
6.1 Sprint 1. Instalación Servidor 2 31

6.1.1 Requisitos De Software. 31
6.1.2 Instalación DSpace. 34
6.1.3 Implementación PID por GRNET. 35
6.1.4 Esquema De Metadatos. 36
6.2 Sprint 2. Ingestión De Archivos Usando SAF 39
6.2.1 SAF. 39
6.3 Sprint 3. Descargas Optimizadas 40
6.4 Sprint 4. Módulo Cargas Web 42
6.4.1 Django-Filer. 43
6.4.2 Ingestión a DSpace. 44
6.5 Sprint 5. Protocolo OAI-PMH 44
6.5.1 OAI-PMH Servidor. 44
6.5.2 OAI-PMH Cliente. 45
6.6 Sprint 6. Optimizar Instalación Y Despliegue 46
6.6.1 Desplegar servicio PID. 47
6.6.2 Módulo Cargas Django Filer. 47
6.6.3 Protocolo OAI-PMH. 47
6.6.4 Máquina Virtual. 48

7 Conclusiones **49**

8 Recomendaciones **51**

Bibliografía **52**

Lista de figuras

Figura 1 Uso de software de código abierto en el mundo para repositorios. 20
Figura 2 Arquitectura DSpace. 22
Figura 3 Diagrama general DSpace. 23
Figura 4 Diagrama del funcionamiento del protocolo OAI-PMH. 25
Figura 5 Diagrama de funcionamiento del servicio de Identificadores Persistentes ofrecido por GRNet. 26
Figura 6 Interfaz JSPUI de DSpace 35
Figura 7 Captura de pantalla con ejemplo de identificador persistente en lagoproject.uis.edu.co 37
Figura 8 Captura de pantalla interfaz de administración de esquemas de metadatos en lagoproject.uis.edu.co 38
Figura 9 Árbol de directorios SAF 41
Figura 10 Captura de pantalla interfaz de descargas desde resultados de búsqueda en racimo.uis.edu.co 42
Figura 11 Captura de pantalla interfaz de Administración Django 44
Figura 12 Captura de pantalla interfaz aplicación Filer 45
Figura 13 Interfaz OAI Servidor de DSpace 46

Lista de tablas

Tabla 1 Sistema De Particiones Servidor 2 32

Siglas

Siglas / Abreviaciones

BASH Bourne-again shell

CC Creative Commons

CHAIN − REDS Coordination and Harmonisation of Advanced e-infrastructure for Research and Education Data Sharing

DART Data Accesibility, Reproducibility and Trustworthiness

FRIDA Fondo Regional para la Innovación Digital en América Latina

GNU GNUś Not Unix

HP Hewlett Packard

IP Internet Protocol

JDK Java Development Kit

JSP JavaServer Pages

JSPUI JavaServer Pages User Interface

LAGO LatinAmerican Giant Observatory

MIT Massachusetts Institute of Technology

OAI Open Archives Initiative

PID Persistent IDentifiers

PMH Protocol for Metadata Harvesting

RD Repositorio de Datos

RDF Representational state transfer

RDF Resource Description Framework

RI Repositorio Institucional

SAF Simple Archive Format

SQL Structured Query Language

SWORD Simple Web-service Offering Repository Deposit

TCP Transmission Control Protocol

UIS Universidad Industrial de Santander

USB Universal Serial Bus Definition

VM Virtual Machine

WCD Water Cherenkov Detectors

XML Extensible Markup Language

XMLUI Extensible Markup Language User Interface

Resumen

Título: Implementación Del Repositorio De Datos DSpace De Modo Descentralizado E Incluyendo Protocolos De Descubrimiento Y Trasmisión De Datos[1]

Autor: Alexander Martínez Méndez[2]

Palabras claves: LAGO, Dspace, OAI-PMH, PID, Repositorio,

Descripción:

En este trabajo se describe un conjunto de herramientas y servicios implementados por la colaboración LAGO (Latin American Giant Observatory) para adoptar el concepto DART (Data Accessibility, Reproducibility and Trustworthiness). Metadatos, identificadores persistentes y otras ventajas de la red de repositorios de LAGO son descritos.

LAGO es un observatorio de rayos cósmicos distribuido con detectores cherenkov de agua (WCD) ubicados en diferentes latitudes y latitudes en toda América Latina, esta red ha sido diseñada para medir la evolución temporal del flujo de radiación a nivel del suelo con un alto grado de detalle. Presentamos la estrategia de la colaboración para catalogar y preservar una gran cantidad de datos producidos en los los arreglos extendidos de detectores WCD . El proyecto LAGO está orientado principalmente en realizar investigaciones sobre: El Universo, el Tiempo Espacial y la Radiación Atmosférica.

Dspace, un software de código abierto utilizado generalmente para repositorios institucionales, proporciona funcionalidades básicas para almacenar y recuperar contenidos digitales con una adaptabilidad para tipos no nativos de contenidos y esquemas de metadatos. Es compatible también, con el protocolo OAI-PMH (Open Archive Initiatives Protocol for Metadata Harvesting). Hemos superado una de las limitaciones más importantes de DSpace: su incapacidad para cargar/descargar múltiples registros, desarrollando un script para ingerir datos aprovechando las capacidades de adaptación de DSpace.

A diferencia de otros instrumentos donde los datos fluyen de un solo lugar a un repositorio de datos, en LAGO cada sitio es responsable de conservar y catalogar localmente su datos, generando conjuntos de datos que, sumado a datos producidos durante el análisis y/o simulación de fenómenos de rayos cósmicos representan por detector 1,5 TB / año de datos brutos y 3 TB / año de simulaciones. El servicio "EPIC Api" proporcionado por GRNET es usado para asociar Identificadores Permanentes a cada conjunto de datos.

[1] Trabajo de grado

[2] Facultad De Ingenierías Físico-Mecánicas, Escuela De Ingeniería De Sistemas. Director: Ph. D. Luis A. Núñez

Abstract

Tittle: DSpace Data Repository Deployment in a Decentralized Mode And Including Data Discovery and Transmission Protocols [3]

Author: Alexander Martínez Méndez[4]

Key words: LAGO, Dspace, OAI-PMH, PID, Repositorio,

Description:

In this work we describe a set of tools and services to implement the Data Accessibility, Reproducibility and Trustworthiness (DART) concept in the Latin American Giant Observatory (LAGO) Project. Metadata, Permanent Identifiers and the facilities and other advantages from the LAGO repository network are described in details.

LAGO is an extended cosmic ray observatory with water-Cherenkov (WCD) detectors spanning over different sites at different altitudes and latitudes across Latin America, network designed to measure the temporal evolution of the radiation flux at ground level with extreme detail. We present the LAGO strategy to catalog and preserve a vast amount of data mapping the extended WCD arrangement to a distributed data network. The LAGO project is mainly oriented to perform basic research on: the Extreme Universe, Space Weather, and AtmosphericRadiation.

An open source software (Dspace) which is used generally for institutional repositories, provides basic functionality for storing and retrieving of digital content with a straightforward adaptability for non-native types of contents and metadata schemes. It also supports the OAI-PMH (Open Archive Initiatives Protocol for Metadata Harvesting) protocol. We have overcome one of the most important DSpace limitations: its inability to upload/download multiple records, developing a script to ingest data profiting from the some DSpace capabilities.

Unlike other instruments where data flows from only one place to a data repositories, in LAGO each site preserves and catalogs locally, generating datasets which, in addition, referers not only to raw data but also to data produced during the analysis and/or simulation of cosmic rays phenomena. typically, each detector generates 1.5 TB/year of raw data and 3 TB/year of simulations . We use the "EPIC Api" service provided by GRNET to associate Permanent Identifiers to each dataset.

[3] Bachelor Thesis

[4] Facultad De Ingenierías Físico-Mecánicas, Escuela De Ingeniería De Sistemas. Director: Ph. D. Luis A. Núñez

Introducción

El proyecto LAGO (Latin American Giant Observatory), es una colaboración enfocada en el estudio del universo, clima espacial y radiación atmosférica por medio de arreglos simples de detectores de partículas ubicados en diferentes sitios y altitudes de la geografía LatinoamericanaSidelnik et al. (2015).

A diferencia de la mayoría de experimentos donde los datos generados son centralizados, en LAGO cada sitio se encarga de generar, preservar y catalogar los datos. Estos provienen tanto de los instrumentos como de las simulaciones de fenómenos de rayos cósmicos realizadas por sus investigadores.

DSpace, software libre, usado ampliamente en repositorios institucionales, provee funcionalidades básicas para almacenar y recuperar contenidos digitales. Es un software con grandes características de adaptabilidad a contenidos no nativos y diversos esquemas de metadatos. La implementación de módulos y protocolos como OAI-PMH potencian su funcionamiento y prestaciones.

En este trabajo se presenta la estrategia llevada a cabo por la colaboración para, entre otros, satisfacer una de las necesidades más importantes de la colaboración, tener la posibilidad de cargar y descargar múltiples archivos

Con este se busca también adoptar y dar cumplimiento a la iniciativa DART (Data Accesibility, Reproducibility adn Trustworthiness) creada para garantizar la autoría, confiabilidad, reproducibilidad y accesibilidad de los datos por CHAIN-REDS (Coordination and Harmonisation of Advanced e-infrastructure for Research and Education Data Sharing) un proyecto de comisión Europea orientado par promover y dar soporte tecnológico y científico a diferentes comunidades en el mundo.Barbera et al. (2014)

1 Definición Del Problema

El proyecto LAGO (Latin American Giant Observatory) [Asorey et al. (2015)] es una colaboración de más de 100 investigadores de astropartículas, afiliados a 26 instituciones presentes en 10 países. Este proyecto, motivado por la experiencia del observatorio Pierre Auger [Collaboration et al. (2015)] en Argentina, realiza principalmente investigación en física de altas energías, clima espacial y radiación atmosférica a nivel del suelo.

En la actualidad esta colaboración cuenta con una red de 10 detectores WCD [Sidelnik et al. (2015)](Water Cherenkov Detector) ubicados a diferentes altitudes, desde México hasta la Patagonia, algunos de estos en sitios sin acceso a Internet o telefonía celular. A futuro espera contar con más detectores operativos, entre estos, dos en la península antártica [Sidelnik et al. (2015)][Asorey et al. (2017)].

Para el análisis de los datos generados por los detectores se deben tener en cuenta aspectos atmosféricos como la presión y la temperatura ambiente, para esto, los detectores están equipados con una cantidad importante de sensores ambientales. El registro, preservación y diseminación de estos datos climáticos, capturados a distintos pisos térmicos a lo largo del continente americano, representa una oportunidad para otras comunidades como los ecologistas quienes podrían hacer uso de ellos para el estudio de los efectos del calentamiento global a gran escala en nuestro continente.

Cada uno de estos detectores registra datasets cada hora con los que aproximadamente se generan 150 GB de datos al mes, obteniendo un total de 1.5 TB/mes de datos de toda la red. Adicional a esto, con el uso de software especializado como lo es CORSIKA [Heck et al. (1998)] (COsmic Ray SImulations for KAscade), ROOT, GEANT4, entre otros, se genera una cantidad importante de datos a partir de simulaciones detalladas de lluvias de partículas provenientes de rayos cósmicos de altas energías.

LAGO, respaldando las iniciativas de acceso libre al conocimiento, decidió utilizar el software DSpace [Team (2016)] para crear un repositorio en el que pudiese almacenar los datos y sus distintas publicaciones académicas. Esta elección se hizo teniendo en cuenta las siguientes características de DSpace: un software de código abierto de distribución gratuita, adaptable a las necesidades de la colaboración; es uno de los software más completo en el mercado; soporta el esquema de

metadatos Dublin Core; soporta los protocolos de interoperabilidad OAI-PMH [Lagoze et al. (2015)] y SWORD; cuenta con una gran comunidad para su mantenimiento; soporta la integración de servicio de Identificadores Digitales Persistentes (PID, por sus siglas en inglés).

DSpace sin duda es uno de los software más completo del mercado, sin embargo, para la colaboración aún quedaban necesidades por cubrir, y sin las cuales la acogida en la colaboración no sería satisfactoria. En primer lugar, se hacía necesario agilizar el proceso de descarga de archivos, generalmente en DSpace se debe ir a cada item para poder descargar el archivo; el proceso de carga de archivos desde la interfaz web necesitaba contar con una alternativa mucho más ágil, especialmente por el hecho de que los metadatos de cada archivo deben ingresarse manualmente, el panorama se complica cuando el detector se encuentra en un sitio sin conectividad y se debe subir un mes de archivos; la implementación del protocolo OAI-PMH era otra de las necesidades por cubrir, con este es posible, entre otros, consultar desde un solo repositorio items de uno o más repositorios; finalmente y no menos importante, disminuir el tiempo y recursos en la instalación y despliegue del repositorio para X sitio era otra de las necesidades por cubrir.

Es en este escenario en el que surge la necesidad de desarrollar este proyecto para buscar la solución a lo mencionado anteriormente para así, lograr una mejor acogida del repositorio como y un mejor escenario en los procesos de investigación de la colaboración.

2 Estado Del Arte

2.1 Repositorios Digitales

Desde una perspectiva general, los repositorios digitales se pueden clasificar en [Luis Alejandro Torres Niño ; Directores Luis A. Núñez (2016)]:

- Repositorios Institucionales (RI): Destinados a ingerir contenidos digitales, de cualquier tipo, generados por una institución.
- Repositorios temáticos: enfocados en un tema particular, generalmente extendidos a varias instituciones.
- Repositorios de formato: estos, se enfocan en un tipo de contenido, entre estos, podemos encontrar: datos de investigación, imágenes digitales, tesis, etc.

En un estudio de comparación de repositorios de datos realizado por la universidad de Minnesota, EEUU [Johnston et al. (2016)], han tomado 6 repositorios de diferentes centros académicos y de estos, 4 utilizan software para repositorios institucionales.

Los repositorios han surgido de la necesidad de grupos específicos de preservar, facilitar la difusión de los datos obtenidos por sus instrumentos o procesos computacionales y facilitar la reproducibilidad dando mayor credibilidad a los resultados obtenidos [Torres et al. (2011)]. En general, encontramos aspectos en común como los son: Datos abiertos; datos fácilmente identificables por sus metadatos; y finalmente, Datos con identidad gracias al uso de identificadores digitales persistentes (PID). Aspectos que se resumen en escalabilidad, integridad y accesibilidad [Luis Alejandro Torres Niño ; Directores Luis A. Núñez (2016)].

Las aplicaciones desarrolladas a la medida para repositorios de datos generalmente se quedan para el uso exclusivo de la comunidad que la desarrollo y las que están en el mercado tienen altos costos, por estas razones, con el ánimo de no "reinventar la rueda" y también teniendo en cuenta el factor económico, LAGO, encontró en los repositorios institucionales su mejor opción, siendo

éstos mucho más generalizados, el software DSpace, un software de código abierto, de distribución gratuita y con muchas de las características de un software para repositorios de datos.

3 Objetivos

3.1 Objetivo General

Disponer de una red de repositorios funcional[1], descentralizada en la cual se pueda hacer búsquedas consolidadas y recuperación distribuida de grandes volúmenes de datos.

3.2 Objetivos Específicos

En busca de convertir un repositorio institucional en una herramienta de uso diario, que facilite el trabajo tanto de investigadores como de administradores de sistemas, se plantean los siguientes objetivos específicos:

1. Desarrollar un módulo para DSpace que permita descargas por lotes y desde múltiples archivos de datos provenientes de búsquedas consolidadas en la red de repositorios.
2. Desarrollar un módulo para DSpace que permita hacer cargas locales a través de dos esquemas complementarios:
   (a) Desde múltiples archivos de datos con la incorporación automática de metadatos mediante el protocolo SWORD y la autoría a través de un servicio PID.
   (b) Desde una interfaz web a partir de cualquier tipo de dispositivo digital.
3. Implementar el protocolo OAI-PMH para la recolección de metadatos desde los demás repositorios de la colaboración LAGO.
4. Desarrollar un script en bash que y crear una máquina virtual que posibiliten la instalación y despliegue de Dspace con los módulos de los literales anteriores y las dependencias de software que se requieran.

[1] Funcional, en el contexto de este proyecto, hace referencia al hecho de que la red de repositorios se convierta en una aliada para cada uno de los investigadores, que a partir del uso de esta los procesos de investigación y sus resultados, sean mas ágiles y de mejor calidad.

4 Marco Teórico

4.1 Repositorio Institucional

Para Clifford Lynch un repositorio institucional (RI) es un conjunto de servicios para almacenar y hacer accesibles materiales de investigación en formato digital creados por una institución y su comunidad, una colección digital del producto de la investigación llevada a cabo por esa comunidad. Ellos pueden formar parte de un sistema mayor, nacional, regional y global de repositorios, indizados de una manera estándar y recuperable, utilizando una interfaz de acceso [Lynch (2003)].

Así, un repositorio institucional es una herramienta muy valiosa para preservar y dar acceso abierto a los contenidos digitales de una comunidad. Fomentar su uso permite recopilar de manera mucho más ágil los contenidos generados y su difusión.

4.2 DSpace

Dspace es un software de código abierto que permite al usuario conservar y acceder de manera fácil y abierta a una gran variedad de contenidos digitales, entre estos, texto, imágenes, libros, vídeos, y datasets. Esta herramienta ha sido desarrollada en sus inicios por el Massachusetts Institute of Technology (MIT) y los laboratorios de Hewlett Packard Company (HP), en la actualidad voluntarios contribuyen en su desarrollo y mantenimiento.[Asorey et al. (2015)][Barbera et al. (2014)][Torres et al. (2011)]

DSpace es el software más utilizado para la creación de repositorios institucionales, en la base de datos de repositorios de código abierto OpenDOAR DSpace se encuentra instalado en el 44.3% de estos repositorios [ver Figura 3.1]. Además de su capacidad para adaptarse a las necesidades de los usuarios al ser un software de código abierto, el soporte a servicios de identificadores persistentes digitales (PID, por sus siglas en inglés) y el soporte a protocolos de transmisión y descubrimiento de metadatos, han favorecido su crecimiento y popularidad en el medio.

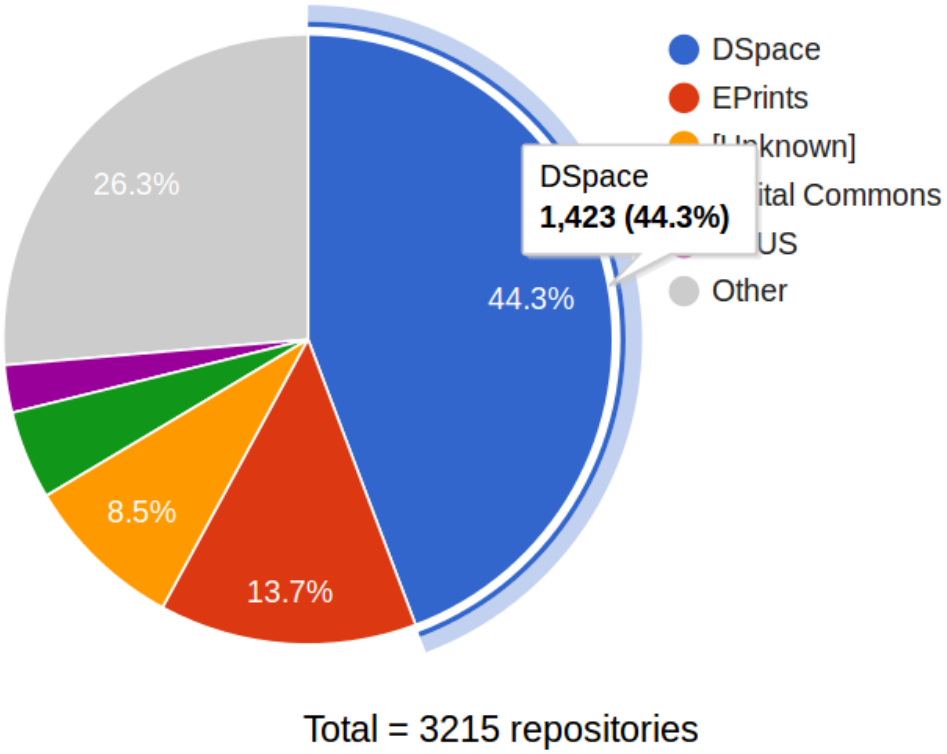

Figura 1 Uso de software de código abierto en el mundo para repositorios.
Adaptado de OpenDOAR, base de datos de repositorios. opendoar.com

Dspace en su arquitectura se encuentra organizado en tres capas, ver figura 2, cada una de estas compuesta por componentes. La capa de almacenamiento (Storage Layer) se encarga del almacenamiento de los contenidos y metadatos. La capa Lógica (Business Logic Layer) se encarga de la gestión de los contenidos en archivo, los usuarios del repositorio (e-people), de los permisos y el flujo de trabajo. La capa de aplicación (Application Layer) contiene componentes para la comunicación con el usuario final a través de las siguientes aplicaciones:

- xmlui
  Interfaz de usuario basada en XML.
- jspui
  Interfaz de usuario basada en JSP.
- solr
  Aplicación web de Apache Solr, utilizada por "xmlui" y "jspui" para la funcionalidad de búsqueda y exploración.

- oai
  Interfaz de la implementación del protocolo OAI-PMH, permite la recopilación de Metadatos y Bitstream.
- rdf
  Interfaz de RDF de DSpace, esta soporta datos vinculados abiertos.
- rest
  API de DSpace REST
- Sword
  Interfaz de DSpace SWORDv1.
- Swordv2
  Interfaz de DSpace SWORDv2.

Cada capa invoca únicamente a una capa inferior. Los componentes de las capas de almacenamiento y lógica se encuentran definidos en un API público por capa, denominándose Storage API y Dspace Public API respectivamente. Estos API's se encuentran desarrolladas en lenguaje JAVA utilizando métodos, clases y objetos.

Aunque la autorización para la ejecución de acciones se da en la capa Lógica, la seguridad del sistema se basa en la correcta y segura autenticación de los usuarios por las aplicaciones en la capa de aplicación.

Funcionalmente DSpace inicia con un proceso de ingestión de archivos con sus metadatos, este conjunto compone un ítem, jerárquicamente cada ítem pertenece a una colección y cada colección a una comunidad. La gestión de items, colecciones y comunidades es realizada por los usuarios de acuerdo con el sistema de permisos establecido. El usuario final puede consultar o buscar los archivos a través de una interfaz web. Ver figura 3

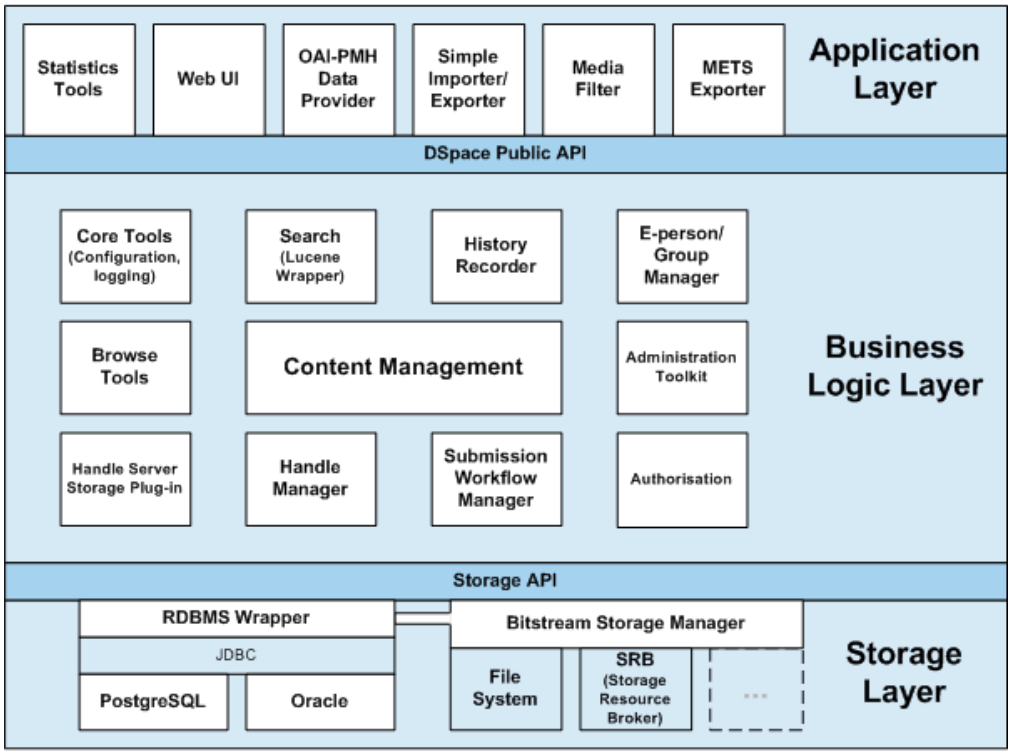

Figura 2 Arquitectura DSpace.
Adaptado de documentación oficial DSpace
https://wiki.duraspace.org/display/DSDOC4x/Architecture

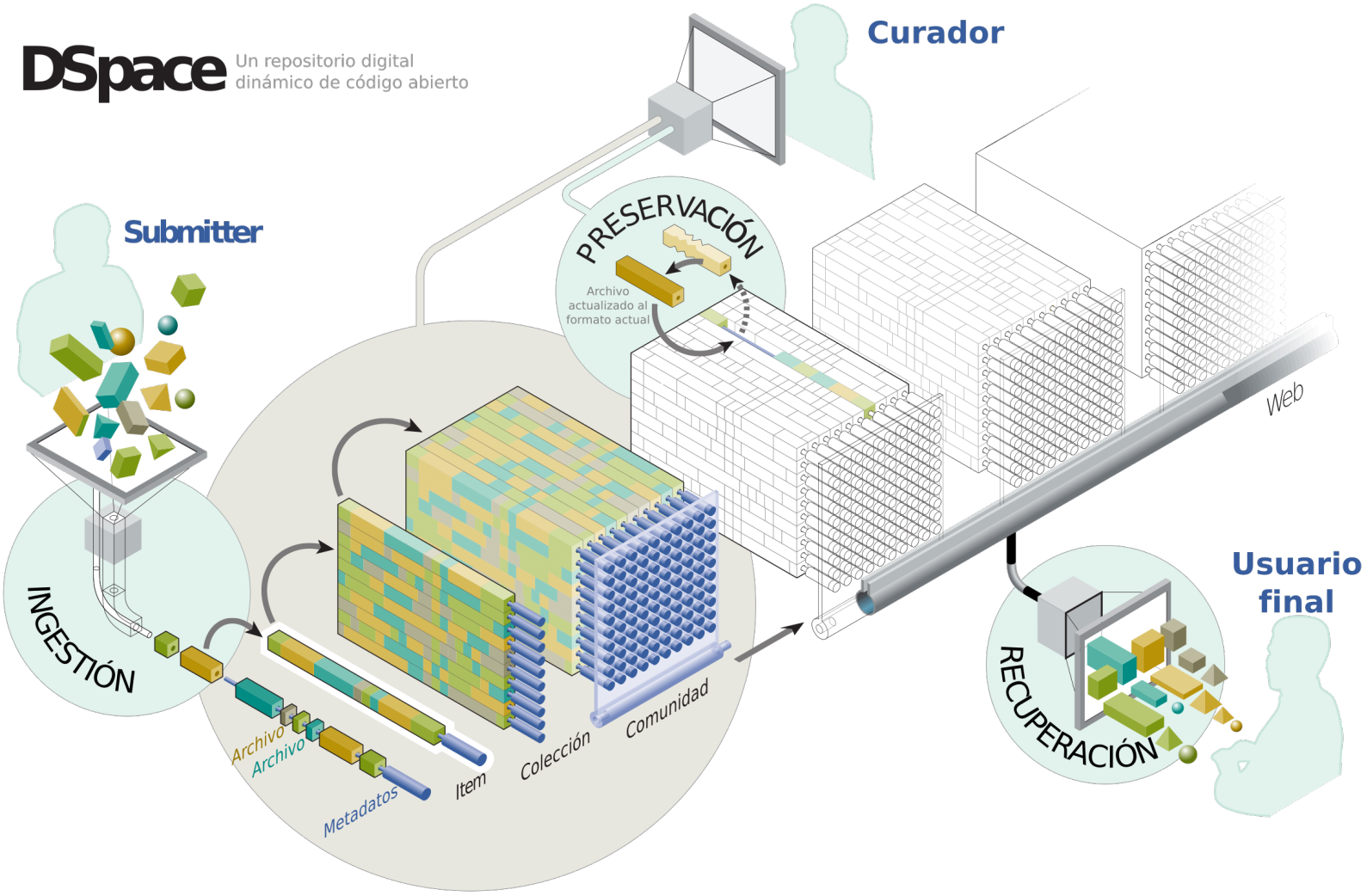

Figura 3 Diagrama general DSpace.
Adaptado de documentación oficial DSpace https://wiki.duraspace.org/display/DSDOC6x/DSpace+6.x+Documentation

4.3 Protocolos De Transmisión Y Descubrimiento De Datos

La ciencia y la industria en la actualidad han desarrollado instrumentos con la capacidad de producir cantidades exorbitantes de datos. Ya sea para datos en crudo o con algún tipo de preprocesamiento, es de vital importancia tener acceso confiable a éstos, en este escenario, los protocolos de transmisión y descubrimiento de datos juegan un rol determinante, acceso rápido, confiabilidad de los datos y el uso de un lenguaje común son indicadores a tener en cuenta en los procesos de investigación o producción.

A continuación, se enunciarán algunos de los protocolos de relevancia en el desarrollo de este proyecto.

4.3.1 TCP/IP. El conjunto de protocolos TCP/IP permite a computadores, teléfonos inteligentes o cualquier dispositivo conectado a una red de área amplia (WAN, por sus siglas en inglés) comunicarse entre sí, en él se especifica como los datos deben ser empaquetados, dirigidos, transmitidos, enrutados y recibidos. Es la base de lo que comúnmente llamamos Internet. [Fall and Stevens (2011)] Considerado un protocolo abierto al tener disponibles públicamente, bajo ningún o muy poco las definiciones de su sistema y gran parte de sus derivados. TCPIP definió capas para las facetas más relevantes en el proceso de comunicación. El modelo OSI (Open System Interconecction) definido por la ISO (International Organization for Standarization) es el modelo de capas de protocolo más utilizado [Fall and Stevens (2011)]

Ya sea por problemas de hardware, radiación, rango de la red u otros factores, es probable que los datos transmitidos sufran perdida o daño de información, TCPIP se encarga del control de estos errores ya sea usando algoritmos matemáticos de recuperación o simplemente retransmitiendo los datos.[Fall and Stevens (2011)]

4.3.2 OAI-PMH. El protocolo OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) proporciona un marco independiente de interoperabilidad basado en la recopilación de metadatos[Lagoze et al. (2015)]. En el marco de OAI-PMH existen dos actores:

- Los Proveedores de datos (Data Provider) quienes exponen metadatos estructurados a través del protocolo OAI-PMH; y
- Los Proveedores de Servicios (Service Provider) quienes utilizan el protocolo OAI-PMH para recolectar estos metadatos.

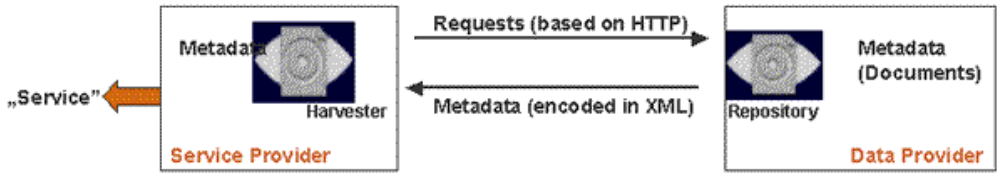

Figura 4 Diagrama del funcionamiento del protocolo OAI-PMH.
Adaptado de Open Archives Forum http://www.oaforum.org/tutorial/english/page3.htm

OAI-PMH es un conjunto de seis verbos invocados a través de peticiones HTTP, los cuales permiten el depósito e intercambio de datos y metadatos entre repositorios, estos son:

1. Identify
   Verbo utilizado para obtener información de un repositorio y así poder identificarlo.
2. ListMetadataFormats
   Utilizado para obtener los formatos de metadatos de un repositorio.
3. ListSets
   Utilizado para obtener la estructura de conjunto de un repositorio.
4. ListIdentifiers
   Abreviación de ListRecords con el cual obtenemos las cabeceras en lugar de los registros.
5. ListRecords
   Utilizando este verbo se recolectan los registros de un repositorio.
6. GetRecord
   Utilizado para obtener un registro de metadatos individual desde un repositorio.

4.3.3 SWORD. SWORD (Simple Web-service Offering Repository Deposit) desarrollado en 2007, es un protocolo ligero que permite el depósito de contenidos de una ubicación a otra, se enfoca especialmente en el depósito de contenidos entre repositorios. También puede ser usado para el depósito de contenidos en cualquier sistema que esté dispuesto a recibirlos. En 2011 se liberó la versión 2, en ésta se extendió el soporte a AtomPub logrando un apoyo completo al ciclo de vida de los depósitos. [Lewis et al. (2012)]

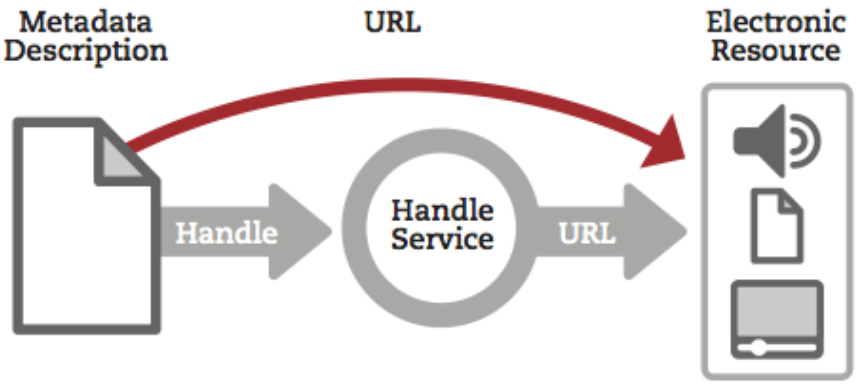

Figura 5 Diagrama de funcionamiento del servicio de Identificadores Persistentes ofrecido por GRNet.

Adaptado de CLARIN http://www.clarin.eu

4.4 PID (Persistent Identifiers)

Los PID´s o identificadores persistentes para objetos digitales permiten entre otras cosas: reclamar los créditos apropiados y eliminar ambigüedades en la identificación y evaluación de publicaciones; lidiar con el problema de un posible cambio de nombre del autor; vincular la producción científica con los contribuidores del proyecto; destacar el trabajo realizado por un investigador en proyectos de equipos grandes. La cantidad de contenidos digitales ha llevado a desarrollar sistema de identificación más completos comparados a los tradicionales ISSN (International Standard Serial Number) e ISBN (International Standard Book Number) [Hakala et al. (2010)]. DOI, PURL, URN y PID-GRNET son algunos de los servicios desarrollados para suplir dicha necesidad. En este proyecto se ha utilizado el servicio de identificadores proporcionado por la organización griega GRNet [Asorey et al. (2015)].

4.5 WCD (Water Cherenkov Detectors)

Los detectores cherenkov de agua (WCD) se basan en el efecto cherenkov. Un tanque hermético a la luz construido ya sea de plástico, fibra de vidrio, metal u otro material se llena con agua y dentro de este un fotomultiplicador se encarga de captar la energía generada por una partícula al cruzar el volumen de agua, la electrónica del tanque se encarga de recibir las señales y transmitirlas a un equipo de cómputo a través de una interfaz USB. [Sidelnik et al. (2015)]

4.6 MetaDatos

Datos sobre otros datos, los metadatos se encargan de describir un archivo facilitando la identificación y búsquedas de contenidos. Existen diversos esquemas de metadatos, esquemas estándar como es el caso de Dublin Core y esquemas únicos como el utilizado por la colaboración LAGO para los datos generados por sus instrumentos en las actividades de investigación [Asorey et al. (2015)].

4.6.1 Dublin Core. Desarrollado y patrocinado por DCMI (Dublin Core Metadata Initiative), Dublin Core es un estándar para la descripción de recursos en dominios cruzados. la implementación de este se hace bajo XML (Extensible Markup Language) y permite definir elementos de un archivo como Título, Asunto, Descripción, Fuente, Idioma, Relación, Cobertura, creador, editor, colaborador, Derechos, Fecha, Tipo, Formato e Identificador.

4.7 Virtualización

La evolución en las prestaciones del hardware en computadores ha permitido a la tecnológica de la virtualización tener un papel protagónico en la computación moderna. La virtualización a grandes rasgos nos permite tener en un solo equipo real varias máquinas virtuales "VM" (Virtual Machine), cada una con su propio sistema operativo [Barham et al. (2003)]. Esta tecnología nos permite, entre otros, dar independencia a los servicios ofrecidos, creación de laboratorios virtuales en ambientes académicos, ofrecer recursos de cómputo de acuerdo a las necesidades del usuario y facilitar los procesos de respaldo y migración.

4.7.1 Hipervisores. Con la eficiencia como pilar fundamental de la virtualización surgen los Hipervisores (Hypervisors's) como medio para garantizar una distribución transparente de los recursos. Un Hipervisor es un software que al estar instalado en el computador anfitrión (Host Computer) controla la forma en que los recursos físicos son usados por la máquinas virtuales. A continuación, se citan algunos tipos de hipervisores:

Hipervisor De Tipo 1. Estos se ejecutan directamente en un anfitrión físico, controlan el hardware y gestionan los sistemas operativos huéspedes. Las máquinas virtuales son independientes, pero comparten los mismo recueros de red.

Hipervisor De Tipo 2. Los hipervisores de tipo 2 se ejecutan sobre un sistema operativo ya sea Windows, GNU/Linux, etc. El sistema operativo huésped tiene acceso a los recursos de red, hardware y gestiona las conexiones con estos recursos. El hipervisor se encarga de controlar los llamados de las máquinas virtuales a los recursos que necesita, CPU, memoria, almacenamiento o red.

4.8 Demonio Cron

El demonio cron es un paquete incluido en la mayoría de las distribuciones GNU/Linux y se encarga de controlar los trabajos que ocurren regularmente, este se activa cada minuto y realiza las acciones establecidas. [Keller (1999)]

4.8.1 Uso Del Demonio Cron.

Cada usuario puede hacer uso de esta herramienta mediante un archivo en el que se especifican las tareas y el tiempo en el que se deben realizar. Con el siguiente comando se realiza la edición del archivo en mención.

```
crontab -e
```

Se selecciona un editor y finalmente se programan las tareas. Para programar una tarea se debe insertar una línea con el siguiente formato:

m h dom mon dow **command**

Donde:

m Minuto de la hora [0-59]
h Hora del día [0-23]
dom Día del mes.
mon Mes del año [1-12]
dow Día de la semana [1-7] 1=lunes, 7=domingo.
command Comando a ejecutarse

Se puede usar "*" para definir "cualquier". Por ejemplo, con la siguiente línea se realiza un backup del directorio home todos los lunes del año a las 05:00 horas.

```
0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
```

5 Metodología

Existe una gran variedad de metodologías para el desarrollo de proyectos, algunas de carácter general y otras especializadas en un tipo de proyecto. Para el desarrollo de éste proyecto se utilizó la metodología Scrum [Schwaber and Sutherland (2011)] teniendo en cuenta que el objetivo principal del proyecto podía cumplirse de modo incremental, obteniendo en periodos cortos de tiempo incrementos terminados del producto, utilizables y potencialmente desplegables. Además, ésta metodología ofrece grandes ventajas para equipos multidisciplinarios y en nuestro caso, trabajar con investigadores de la colaboración LAGO ha sido de gran utilidad para comprender correctamente sus necesidades.

El desarrollo del producto final se realizó en seis (6) sprints, cada uno de estos inició con la reunión de planificación y finalizó con la entrega del incremento completamente funcional, en ese intervalo se realizaron reuniones periódicas donde se inspeccionaba el desarrollo del incremento, adaptando la lista de producto cuando fuese necesario. A continuación una descripción de los sprints.

- Sprint 1
  En el primer Sprint de este proyecto se realizó la instalación de DSpace en un segundo servidor.
- Sprint 2
  En este se desarrolló un módulo de ingestión de archivos al repositorio mediante scripts desarrollados en python y bacth, estos esencialmente hacen uso de la herramienta rsync y el método de ingestión SAF (Simple Archivo Format) de DSpace.
- Sprint 3
  Se modificó el código fuente de DSpace de tal manera que permitiese realizar descargas múltiples a partir de los resultados en los procesos de búsqueda.
- Sprint 4
  Se puso al servicio un módulo para carga de múltiples archivos al servidor y posterior ingestión al repositorio.

- Sprint 5
  Se implemento el protocolo OAI-PMH para permitir la difusión e ingestión de metadatos entre repositorios

- Sprint 6
  En el último Sprint se crearon ejecutables para facilitar el proceso de instalación y despliegue de un repositorio para cualquier sitio de la colaboración LAGO.

6 Desarrollo Del Proyecto

En este capítulo se presenta una descripción de cada una de las etapas (sprints) en el desarrollo de este proyecto.

6.1 Sprint 1. Instalación Servidor 2

Con el objetivo de verificar la funcionalidad de cada uno de los desarrollos y evitando posibles fallas en el servicio ofrecido en lagoproject.uis.edu.co se realizó la instalación de Dspace en un segundo servidor, equipo con las siguientes características:

- Sistema operativo: CentOS 7.0
- Procesador: 2 núcleos
- Memoria RAM: 14 GB
- Almacenamiento: 2 TB

En http://racimo.uis.edu.co:8080/repository/ se puede acceder a esta instalación. El equipo en el que se encuentra instalado hace parte del proyecto Racimo, ejecutado por el grupo Halley de Astronomía y Ciencias Aeroespaciales miembro de la colaboración LAGO y adscrito a la escuela de Física de la Universidad Industrial de Santander, UIS. Este proyecto ha sido financiado por el programa FRIDA (Fondo Regional para la Innovación Digital en América Latina) y la Vicerrectoría de Investigación de la Universidad Industrial de Santander, UIS.

6.1.1 Requisitos De Software.

Para la instalación de DSpace 6.x (Lanzamiento actual) se requiere del siguiente software:

Sistema Operativo. Se necesita un sistema operativo ya sea de tipo UNIX o Microsoft Windows, para este proyecto se partió de una instalación base del sistema operativo CentOS en su versión 7.0 con el siguiente sistema de particiones:

Tabla 1 Sistema De Particiones Servidor 2

Partición	Punto De Montaje	Tamaño
/dev/sda1	/	29 GB
devtmpfs	/dev	6.8 GB
tmpfs	/dev/shm	6.8 GB
tmpfs	/run	6.8 GB
tmpfs	/sys/fs/cgroup	6.8 GB
/dev/sdd1	/datadrive1	985 GB
/dev/sdc1	/datadrive	985 GB
/dev/sdb1	/mnt/resource	133 GB

OpenJDK. Se requiere de la versión 7 o superior del kit de desarrollo de java, edición estándar, este puede ser descargado de la dirección: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Pasos de instalación:

1. Descargar el paquete jdk-8u131-linux-x64.tar.gz
2. Desempaquetar el software
3. Agregar las variables de entorno
4. Actualizar las alternativas para java
5. Actualizar el entorno

Apache Maven. Este software es necesario en la primera etapa del proceso de compilación para montar el paquete de instalación. Se requiere de la versión 3.0.5 o posterior. Dirección de descarga https://maven.apache.or
g/download.cgi

Pasos de instalación:

1. Descargar el paquete apache-maven-3.3.9-bin.tar.gz
2. Desempaquetar
3. Agregar las variables de entorno
4. Actualizar el entorno

Apache Ant. Software necesario en la segunda etapa del proceso de compilación en primer lugar para construir el instalador y en segunda instancia para desplegar DSpace al directorio de instalación. Se requiere de la versión 1.8 o posterior. Dirección de descarga http://ant.apache.org/bindownload.cgi

Pasos de instalación:

1. Descargar el paquete apache-ant-1.8-bin.tar.gz
2. Desempaquetar
3. Agregar las variables de entorno
4. Actualizar el entorno

Base de Datos Relacional. Dspace puede utilizar como motor de base de datos PostgreSQL 9.4 o posterior u Oracle 10 o posterior, para este proyecto se utilizó PostgreSQL en su versión 9.4 y su instalación se realizó a través de los repositorios oficiales.

Pasos de instalación:

1. Instalar los paquetes postgresql-9.4 y postgresql-contrib-9.4
2. Asignar contraseña al usuario "postgres"
3. Iniciar sesión como usuario "postgres" y modificar la contraseña del usuario "postgres" en la consola de administración de PostgreSQL
4. Actualizar los puertos de escucha y el método de autentificación del cliente PostgreSQL

Apache Tomcat. Se necesita de Apache Tomcat 7 o posterior para llevar al usuario Dspace a través de un navegador, para este proyecto se utilizó la versión 8 y se instaló desde los repositorios oficiales.

Pasos de instalación:

1. instalar los paquetes: tomcat8 tomcat8-admin tomcat8-docs tomcat8-examples tomcat8-user
2. Crear usuario administrador para Tomcat
3. Garantizar cuotas de memoria para Tomcat
4. Verificar el puerto

6.1.2 Instalación DSpace.

Después de tener el software necesario instalado es posible realizar la instalación de DSpace, la instalación se puede realizar usando los binarios o los archivos fuente, en este caso se utilizó los archivos fuente debido a que es la versión que permite modificar de base el repositorio.

A continuación, un resumen de los pasos de instalación, información más detallada puede ser revisada en la documentación oficial, en el siguiente enlace: https://wiki.duraspace.org/display/DSDOC6x/Installing+DSpace#InstallingDSpace-InstallationInstructions

1. Crear usuario "dspace"
2. Descargar códigos fuente de DSpace
3. Desempaquetar
4. crear usuario y base de datos "dspace" en PostgreSQL
5. Habilitar extensión "pgcrypto" en PostgreSQL
6. Crear archivo de configuración de DSpace, en este se definen parámetros como lo es, el servicio de correo, identificadores persistentes, directorio de instalación, nombre del servicio, etc. Una descripción más detallada de cada uno de los parámetros se encuentra en la documentación oficial [Team (2016)].
7. Crear directorio de instalación
8. Construir el paquete de instalación

9. Instalar DSpace en el directorio de instalación definido

10. Desplegar las aplicaciones web

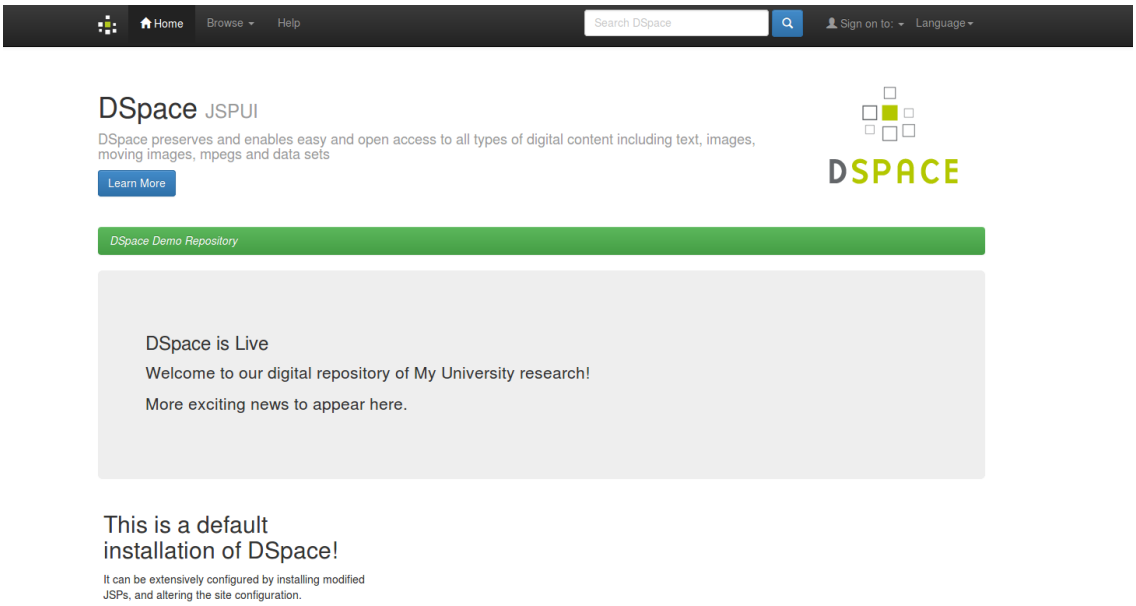

Figura 6 Interfaz JSPUI de DSpace
Captura de pantalla de la interfaz basada en JSP de DSpace, interfaz denominada JSPUI.

6.1.3 Implementación PID por GRNET.

Para la implementación del servicio de identificadores persistentes proporcionado por GRNET es necesario modificar algunos archivos de configuración y también, agregar algunos archivos, el contenido de estos se puede consultar en el repositorio en GitHub en el siguiente enlace halley.uis.edu.co/archivos/dspace. A continuación, el listado de los archivos a agregar y modificar:

- Archivos a modificar:

  - DSpaceSource/dspace/config/dspace.cfg
    En este archivo se modifica el parámetro "handle.prefix" con el prefijo asignado, por ejemplo:
    handleprefix = 11456

- Archivos a agregar:

- DSpaceSource/dspace-4.2-src-release/dspace-api/src/main/java/org/dspace/handle/HandleManager.java
- DSpaceSource/dspace-4.2-src-release/dspace-api/src/main/java/org/dspace/identifier/VersionedHandleIdentifierProvider.java
- DSpaceSource/dspace-4.2-src-release/dspace-api/src/main/java/cz/cuni/mff/ufal/dspace/AbstractPIDService.java
- DSpaceSource/dspace-4.2-src-release/dspace-api/src/main/java/cz/cuni/mff/ufal/dspace/PIDServiceEPICv2.java
- DSpaceSource/dspace-4.2-src-release/dspace-api/src/main/java/cz/cuni/mff/ufal/dspace/PIDService.java
- DSpaceSource/dspace-4.2-src-release/dspace-api/src/main/java/cz/cuni/mff/ufal/dspace/handle/PIDCommunityConfiguration.java
- DSpaceSource/dspace-4.2-src-release/dspace-api/src/main/java/cz/cuni/mff/ufal/dspace/handle/PIDCommunityConfiguration.java

En caso de necesitar hacer pruebas se debe utilizar un usuario de prueba y no el usuario de producción asignado. Teniendo todos los archivos en el código fuente se procede a reconstruir pace, para esto se ejecutan los siguientes comandos y se tendrá el servicio activado y en funcionamiento.

```
cd DSpaceSource
mvn package
cd DSpaceSource/dspace/target/dspace-installer
ant update
systemctl restart tomcat
```

6.1.4 Esquema De Metadatos.

Debido a las particularidades de la colaboración es necesario el uso de un esquema de metadatos propio, el cual debe ser registrado en el repositorio antes de realizar la ingestión de cualquier archivo que tenga este esquema de metadatos. Para esto se debe ingresar a la plataforma de administración web y siguiendo la ruta "Admnister/General Settings/Metdata Registry" crear el esquema de metadatos. En el formulario que se observa en la figura 8 se ingresa el nombre del esquema de metadatos, se guarda y seguido a esto se procede a crear cada campo de metadatos dando clic en el botón "Update".

En la versión 5 (acqua) de los datasets generados por los detectores se optó por un sistema de metadatos en ASC-II plano que son inyectados al repositorio al momento de la adquisición, con este sistema se reúnen los metadatos del dataset y se insertan en un archivo siguiendo un sistema

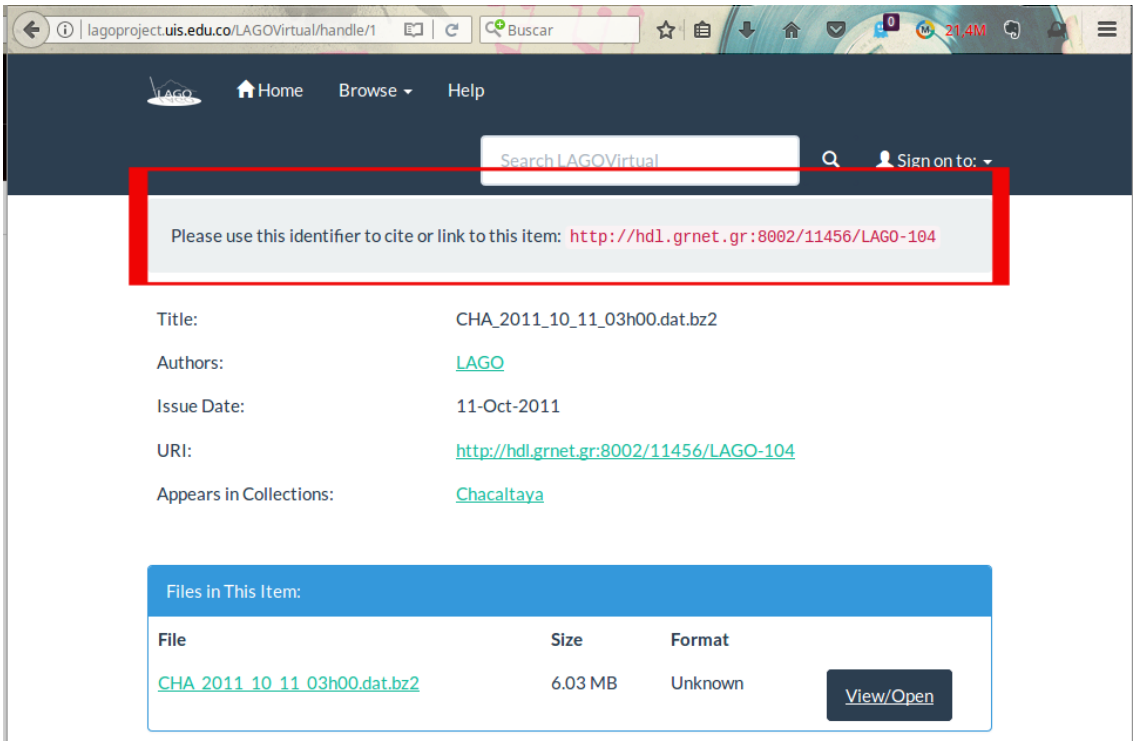

Figura 7 Captura de pantalla con ejemplo de identificador persistente en lagoproject.uis.edu.co

llave-valor, el nombre de estos archivos es de la siguiente forma:

Nombre del dataset. nombreSitio_YYYY_MM_DD_HHh00.dat
Estos usualmente se comprimen en formato bzip2.

Nombre del archivo de metadatos. nombreSitio_YYYY_MM_DD_HHh00.mtd

A continuación una descripción de los metadatos del squema LAGO v5 (acqua):

- Metadatos de configuración (el contenido del archivo lago-configs vigente al momento de la adquisición generado por el script lago-configs.pl)
- Promedio y desvío estándar de la tasa de conteo de pulsos por canal, según el siguiente esquema para las siete posibles combinaciones de disparo:
  - triggerRateAvg1 (0b001): canal 1
  - triggerRateAvg2 (0b010): canal 2

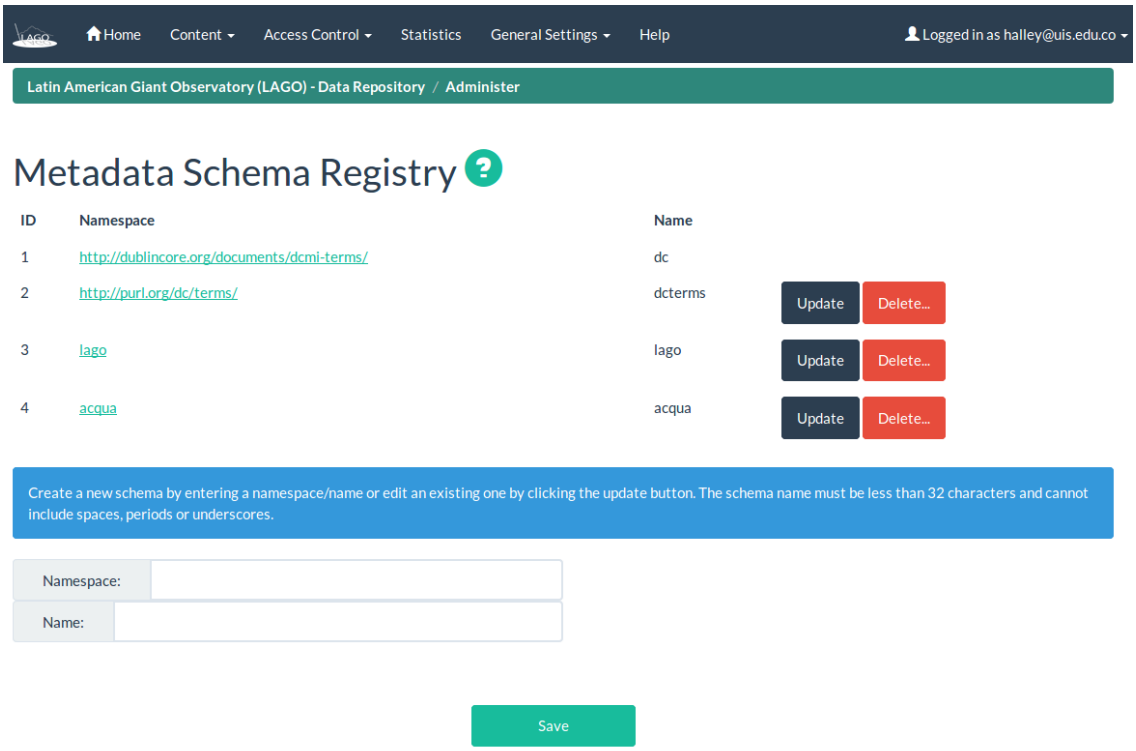

Figura 8 Captura de pantalla interfaz de administración de esquemas de metadatos en lagoproject.uis.edu.co

  - triggerRateAvg4 (0b100): canal 3
  - triggerRateAvg3 (0b011): canales 1 y 2
  - triggerRateAvg5 (0b101): canales 1 y 3
  - triggerRateAvg6 (0b110): canales 2 y 3
  - triggerRateAvg7 (0b111): canales 1, 2 y 3

- Promedio y desvío estándar de la línea de base (baseline) por canal estatus de la adquisición
- Nombre de archivos
- Tiempo de adquisición
- Cantidad de pulsos registrados
- Valor absoluto y fracción del deadtime, definido como el número de pulsos disparados, pero no registrados en el archivo de adquisición. Típicamente, tiene valores < $\tilde{1}0^{\wedge}6$ para tasas de conteo normales, y puede llegar a $\tilde{1}$%-2% para tasas de adquisición máximas ($\tilde{5}0$ kPulsos/s).

En todos los casos, se utiliza el valor -1 para indicar algún fallo en la adquisición.

6.2 Sprint 2. Ingestión De Archivos Usando SAF

En esta etapa del proyecto se desarrolló una serie de scripts para importar datos al repositorio desde los instrumentos, para ésto, un usuario enjaulado transmite los datos al repositorio usando RSync y en el equipo donde está alojado el repositorio diariamente, usando la herramienta cron, se ejecuta un script el cual se encarga de realizar la ingestión usando el método SAF (Simple Archive Format) de DSpace.

6.2.1 SAF. SAF es una herramienta que permite importar o exportar ítems desde la terminal. Con SAF básicamente se crea un árbol de directorios, ver figura 9, en el que tenemos un directorio principal y subdirectorios por cada ítem, en cada subdirectorio se alojan los archivos a importar, archivos xml con los diferentes esquemas de metadatos, un archivos para determinar las colecciones a las que va a pertenecer el ítem y finalmente, un archivo en el que se pone, linea por linea, el nombre de cada uno de los archivos a importar. Teniendo este árbol de directorios se ejecuta la siguiente linea en la terminal desde el usuario de DSpace.

dspacePath/bin/dspace import –add –eperson=joe@user.com –collection=CollectionID
–source=items_dir –mapfile=mapfile

eperson	Correo del usuario en DSpace
collection	El ID de la colección a la que pertenecerán los ítems
source	Ruta en el sistema del árbol de directorios
mapfile	Ruta del archivo mapfile,

En la documentación oficial de DSpace [Team (2016)] se encuentra mayor información sobre este método y sus distintas opciones, ésta se puede consultar en https://wiki.duraspace.org/display/DSDOC5x/Importing+and+Exporting+Items+via+Simple+Archive+Format

A continuación, una descripción del script desarrollado en python para realizar la ingestión de archivos una vez estos sean transferidos al servidor desde los detectores.

Algorithm 1 Ingestión SAF

Require: Ruta binario DSpace, Ruta de los datos, directorio para el "mapfile", email del usuario, ID de Colección
Ensure: Rutas ingresadas validas

```
for archivo in archivos do
    if Versión de archivo igual a LAGO V5 then
        Extraer metadatos LAGO V5 (acqua)
        crear archivo xml de metadatos acqua
    else
        Extraer metadatos LAGO V4
        crear archivo xml de metadatos LAGO V4
    end if
    crear archivo xml de metadatos Dublin Core
    crear archivo "contents" con el nombre de los archivos del item, Dataset y metadatos
    crear archivo "collection" con el ID de la colección donde estará el ítem
    Crear subdirectorio para el ítem
    Mover todos los archivos al subdirectorio
end for
Ejecutar comando de ingestión SAF
```

6.3 Sprint 3. Descargas Optimizadas

Optimizar el proceso de descargas, como se mencionó anteriormente, es clave para la colaboración LAGO, es algo que permite al repositorio ir un paso más allá, no ser únicamente una herramienta de preservación de contenidos sino una herramienta que facilite las actividades que involucren datos de los detectores.

Permitir la descarga de archivos desde los resultados de búsqueda es posible modificando el código fuente de DSpace, para esto, se modifican scripts escritos en java y que utilizan spring, un framework Java de código abierto, desarrollado por SpringSource que proporciona las herramientas necesarias para desarrollar aplicaciones estructuradas, sostenibles y fácilmente comprobables [Sharma and Sarin (2016)].

A continuación, una descripción de los archivos modificados y creados:

- DspaceSource/dspace/config/dspace.cfg
  En este archivo se define un campo a mostrar en los resultados de búsqueda, desde este campo se hace la descarga directa del archivo. El parámetro "webui.itemlist.columns" debe definirse de la siguiente manera:

```
webui.itemlist.columns = mark_availibility, dc.date.issued
```

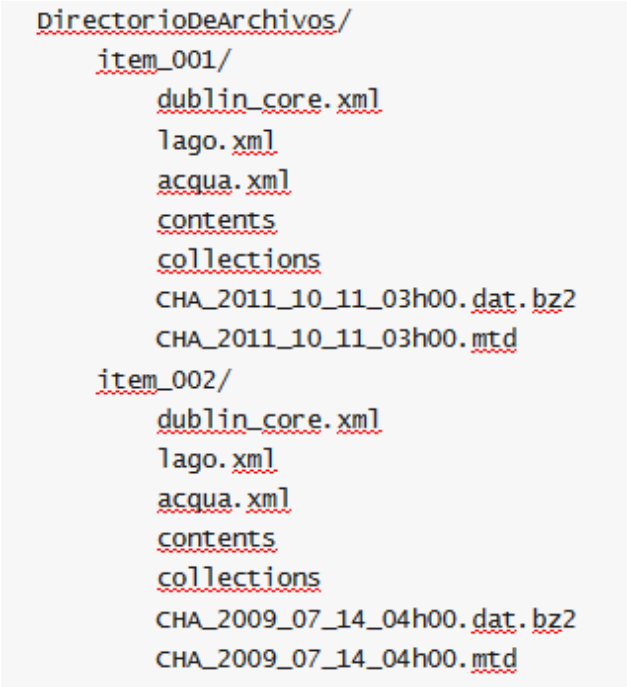

Figura 9 Árbol de directorios SAF
Ejemplo de lo que sería un árbol de directorios para la ingestión de los archivos CHA_2011_10_11_03h00.dat.bz2 y CHA_2011_10_11_03h00.dat.bz2 generados por el detector Chacaltaya de la colaboración LAGO.

```
(date), dc.title, dc.contributor.*
```

- DSpaceSoftware/dspace-api/src/main/resources/Messages.properties En este se define el texto que aparece en el campo creado anteriormente, para esto se modifica el parámetro "item-list.mark_availability" de la siguiente manera:

  ```
  itemlist.mark_availability = Download
  ```

- Ccrear bean Finalmente se crea el bean, archivo donde se dan las instrucciones para la descarga de los archivos, este script se aloja en la ubicación:

  ```
  dspace/config/spring/api/item-marking.xml
  ```

  y su contenido se encuentra en el siguiente enlace: halley.uis.edu.co/archivos/dspace
  En este, se define también la imagen a utilizar para mostrar la disponibilidad del archivo.

En la figura 10 con el recuadro rojo se resalta el espacio para la descarga del archivo, se observa la imagen en verde informando de la disponibilidad de este. El icono utilizado cuenta con licencia Creative Commons CC 3.0 BY, ha sido diseñado por Dave Gandy y se puede descargar de la página web http://www.flaticon.com.

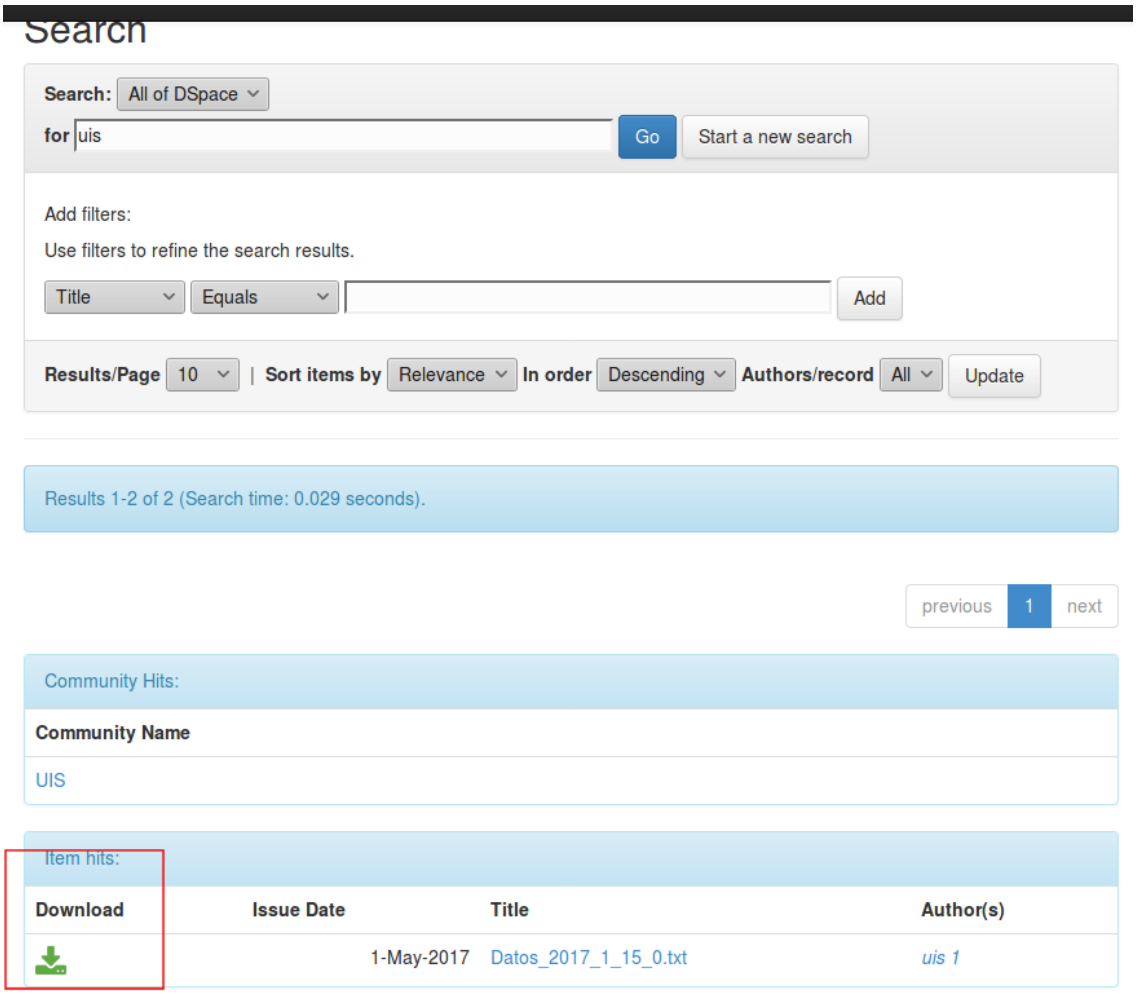

Figura 10 Captura de pantalla interfaz de descargas desde resultados de búsqueda en racimo.uis.edu.co

6.4 Sprint 4. Módulo Cargas Web

En este sprint se puso en servicio un módulo que permitiese a los usuarios importar cantidades considerables de archivos al repositorio desde una interfaz web, esta herramienta es de gran utilidad en escenarios donde los detectores se encuentran en sitios alejados, sin conexión a Internet constante. Para poder publicar los archivos generados por estos detectores, los investigadores deben desplazarse cada determinado tiempo hasta el sitio del detector, allí mediante un disco duro, memoria USB o cualquier medio portable de almacenamiento extraen los archivos para finalmente realizar el proceso de ingestión al repositorio.

Para cumplir con el incremento del sprint, se desplegó la aplicación Django-Filer, aplicación desarrollada con Django, un framework de alto nivel, simple, robusto y flexible desarrollado

en python que permite crear aplicaciones web con relativa rapidez[Forcier et al. (2008)]. Este framework maneja una arquitectura escalable tipo Modelo-Vista-plantilla (MVP) con lo cual nos permite mantener la lógica y el diseño separadas [Adrian Holovaty (2009)].

Adrian Holovaty, Jacob Kaplan-Moss, 2009

Modelo. Capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.

Vista. Capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web o otro tipo de documento.

Plantilla. Capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre modelos y plantillas.

6.4.1 Django-Filer.

Django-Filer es una aplicación con licencia BSD y con el aporte de más de 100 desarrolladores en su repositorio en GitHub, https://github.com/divio/django-filer. Es una aplicación bastante completa para el manejo de archivos y usuarios.

Su despliegue se realizó siguiendo los pasos de la documentación oficial, la cual puede ser consultada en el siguiente enlace https://django-filer.readthedocs.io/en/latest/index.html.

Previamente se realizó la instalación de los paquetes: python 3.4, Django 1.11.3 y virtualenv 15.1.0. Este último es de gran utilidad para solucionar problemas de dependencias de software y/o permisos, con este se crean entornos virtuales independientes [virktualenv Developer Team (2016)].

En la figura 11 observamos la interfaz de administración de Django, en ésta podremos, entre otros, gestionar los archivos con la herramienta Filer y también, crear usuarios, grupos y asignar permisos para el uso de las aplicaciones instaladas.

En la administración de la aplicación Filer podremos asignar permisos a las carpetas si tenemos los privilegios necesarios; podemos gestionar los archivos y carpetas; y finalmente, podremos gestionar el funcionamiento de las vistas en miniatura.

En la interfaz de gestión de archivos y carpetas, ver figura 12, es posible crear, mover o eliminar carpetas; subir, mover, editar o eliminar archivos; también, es posible obtener enlace para compartir estos archivos.

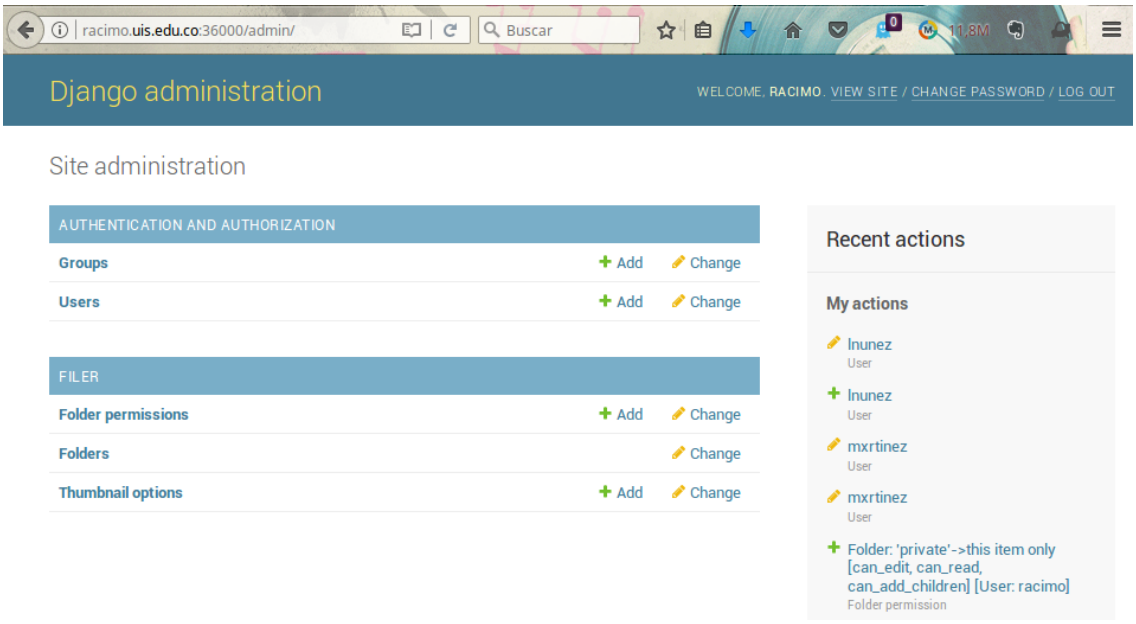

Figura 11 Captura de pantalla interfaz de Administración Django

6.4.2 Ingestión a DSpace. Mediante un script en bash, ejecutado diariamente con cron, se toman los archivos subidos a través de la aplicación Filer, se mueven a un directorio temporal desde donde se realiza la ingestión con el script en python descrito en la página **??**, script que usa el método SAF de DSpace.

6.5 Sprint 5. Protocolo OAI-PMH

El protocolo OAI-PMH es una herramienta de gran utilidad, con la implementación de este podríamos, por ejemplo, obtener datos de Y repositorio en los resultados de búsqueda en X repositorio. La implementación de este protocolo se describe a continuación:

6.5.1 OAI-PMH Servidor. Para habilitar el servidor basta con asegurarse que este desplegada la aplicación DSpacePATH/webapps/oai en el contenedor de servlets(en este caso Tomcat). Para esto, se creó un archivo xml en el que se define un contexto de la aplicación OAI-PMH en tomcat, El contenido del archivo xml es el siguiente:

```
<?xml version='1.0'?>
<Context
    docBase="DSpacePATH/webapps/xmlui"
    reloadable="true"
```

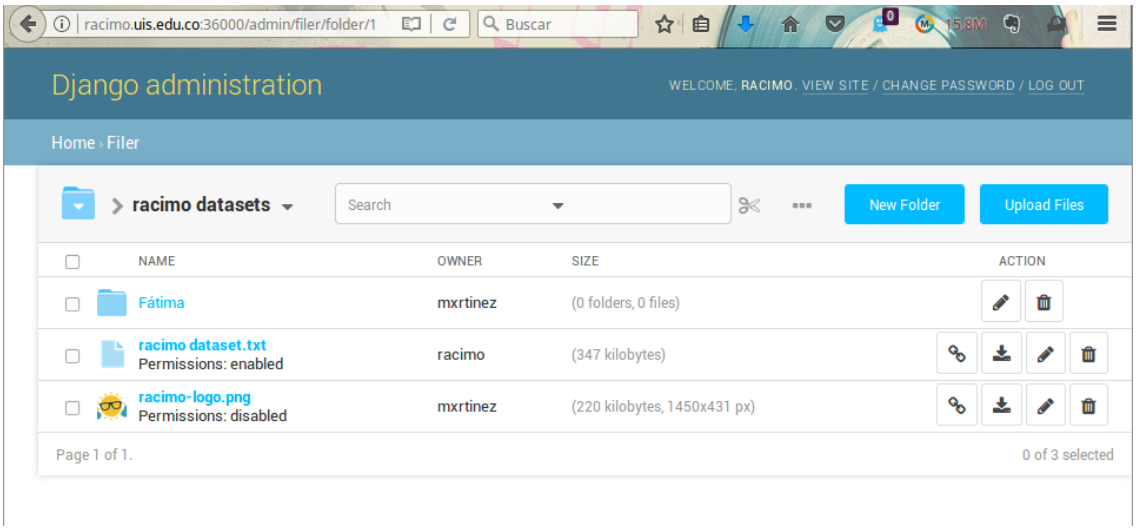

Figura 12 Captura de pantalla interfaz aplicación Filer

```
cachingAllowed="false"/>
```

Es importante mantener actualizado el índice de esta aplicación, para esto es necesario ejecutar, recomendablemente, a diario el siguiente comando:

```
DSpacePATH/bin/dspace oai import -o > /dev/null
```

Esto puede ser realizado de manera automática creando un cron en el que se inserta la siguiente línea:

```
0 0 * * * DSpacePATH/bin/dspace oai import -o > /dev/null
```

Con ésta, el comando se ejecutará todos los días a las 00:00 horas.

La aplicación xmlui de DSpace debe estar instalada para hacer uso de las características del protocolo OAI-PMH.

6.5.2 OAI-PMH Cliente.

Una vez verificado el correcto funcionamiento del servidor OAI-PMH, para realizar la recolección de contenidos basta con establecer los parámetros de configuración del recolector OAI, estos se establecen en el archivo "DSpacePATH/config/modules/oai.cfg" y su descripción se encuentra en la documentación oficial [Team (2016)] en el enlace: https://wiki.duraspace.org/display/DSDOC6x/OAI

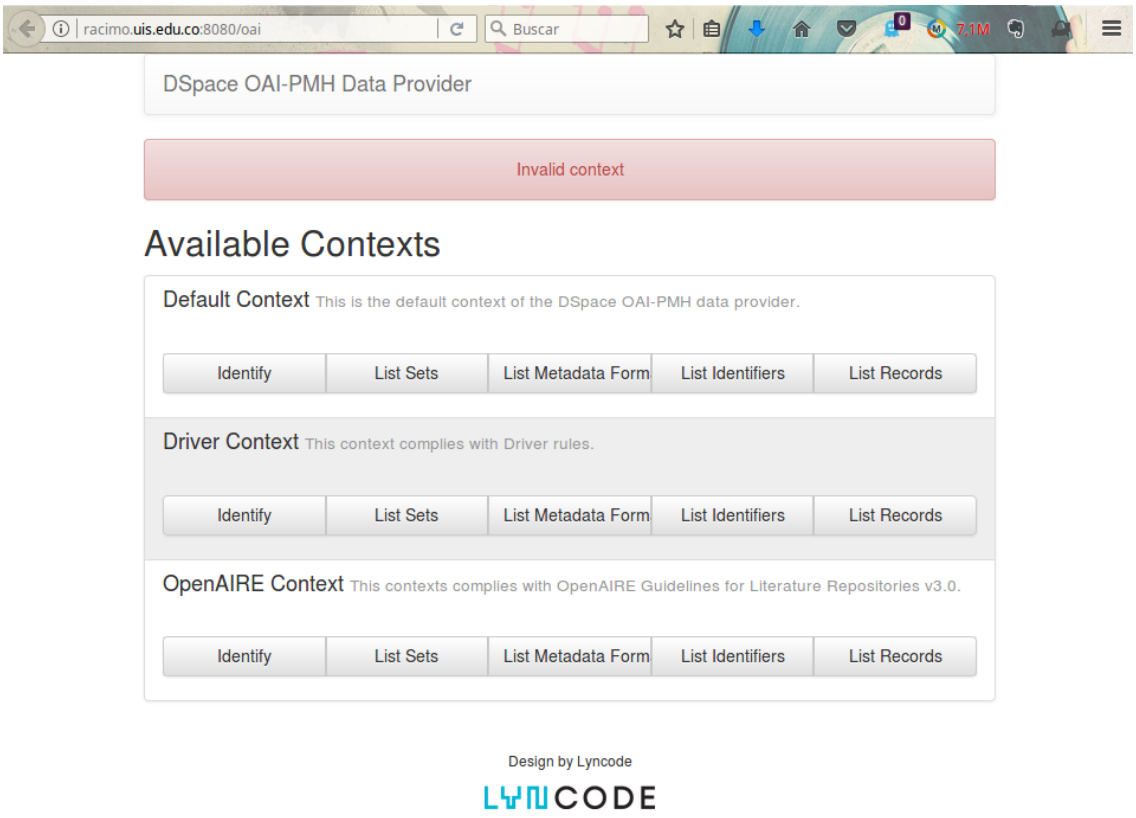

Figura 13 Interfaz OAI Servidor de DSpace
Captura de pantalla de la interfaz de la implementación del protocolo OAI-PMH.

6.6 Sprint 6. Optimizar Instalación Y Despliegue

Teniendo en cuenta la cantidad de instrucciones ejecutadas y archivos modificados en el proceso de despliegue de un repositorio funcional para algún sitio de la colaboración LAGO se crearon scripts en bash o python para optimizar este proceso, con los script se instalan los requisitos de software, se modifican parámetros y se despliegan las aplicaciones de manera más rápida y hasta cierto punto sin la necesidad de un experto. Estos scripts podrán ser consultados en el siguiente enlace halley.uis.edu.co/archivos/dspace.

En el transcurso de cada uno de los sprints anteriores se realizó una documentación de los procesos de relevancia para este sprint, en este sprint se recopilaron todas las instrucciones y se plasmaron en los mencionados scripts.

Para facilitar este proceso y evitar problemas de dependencias o versiones con el software usado, en el siguiente enlace se puede descargar un paquete con los códigos fuente y paquetes binarios.

El código fuente de DSpace descargado desde este enlace es cuanta con todas las modificaciones mencionadas a lo largo de este documento.

http://halley.uis.edu.co/archivos/dspace/

A continuación, las descripción de los demás scripts.

6.6.1 Desplegar servicio PID.

Para agilizar el proceso de implementación del servicio PID proporcionado por GRNET básicamente se copian los archivos del servicio y finalmente se establecen los parámetros del servicio tal y como se menciona en la página 6.1.3.

Algorithm 2 Desplegar servicio PID

Require: handle.prefix, lr.pid.service.user, lr.pid.service.pass

1: Copiar los archivos a las rutas especificadas
2: Definir el valor ingresado de los parámetros en los archivos del servicio PID
3: Reconstruir DSpace

6.6.2 Módulo Cargas Django Filer.

A continuación, la descripción del algoritmo desarrollado para el despliegue de este aplicación:

Algorithm 3 Desplegar aplicación Django Filer

1: Instalar el paquete virtualenv
2: Crear entorno virtual con virtualenv
3: Moverse al entorno virtual
4: Instalar los paquetes python 3.4, Django 1.11.3 y python-pip 3.4
5: Crear proyecto en Django
6: Crear usuario administrador del proyecto
7: Instalar Django Filer con pip
8: Agregar Django Filer al proyecto
9: Desplegar Servicio
10: Crear cron para ingestión SAF

La creación de usuarios a excepción del usuario administrador se hace desde la interfaz de administración web, donde también se establece el sistema de permisos.

6.6.3 Protocolo OAI-PMH.

los parámetros a modificar para la implementación de este protocolo ya se encuentran establecidos en el código Fuente de Dspace disponible en el paquete descargable mencionado en la página 47.

Para agregar en los clientes nuevos servidores OAI-PMH se realiza el procedimiento descrito en la página 6.5.2.

Finalmente, en las siguientes páginas se encuentran descritos los scripts de instalación para: OpenJDK página 32; Apache Maven página 32; Apache Ant página 33; PostgreSQL página 33; Apache Tomcat página 34; y DSpace página 34.

6.6.4 Máquina Virtual. Otra forma de optimizar el proceso de despliegue de un repositorio para la colaboración LAGO es mediante una máquina virtual, en esta tendremos instalado DSpace y bastará con una aplicación que solicite unos parámetros para personalizar el repositorio.

Para crear la máquina virtual si siguió el tutorial ofrecido en la documentación oficial de Debian (https://wiki.debian.org/Xen#Installation_as_a_DomU_.28guest.29), se utilizó un equipo anfitrión con Debian y con Xen como hipervisor.

7 Conclusiones

El mayor logro con el desarrollo de este proyecto ha sido extender la funcionalidad de DSpace. Llevarlo de ser un software para preservación de contenidos, a ser un software de difusión con características de confiabilidad y preservación. Además de contribuir a agilizar las labores de administración del sistema. A continuación una descripción mas detallada de este logro y otros aspectos relevantes en el desarrollo del proyecto.

- El poder descargar archivos directamente desde los resultados de una búsqueda en DSpace, aumentará el uso de esta herramienta y permitirá a más investigadores tener los datos necesarios para sus labores de investigación,
- La aplicación para cargas web implementada es una alternativa mucho más rápida al método tradicional de ingestión desde la interfaz web. Con esta el proceso realizado por el usuario se reduce a autenticarse y arrastrar o seleccionar los archivos.
- la implementación del protocolo OAI-PMH ayuda a la difusión de contenidos y además evita a usuarios ir de repositorio en repositorio para obtener los archivos necesitados.
- El esquema de metadatos definido para los datasets de los detectores mejora, especialmente, los procesos de búsqueda gracias al nivel de detalle con el que se ha definido y la cantidad de metadatos asociados.
- El trabajo realizado para agilizar el proceso de instalación y despliegue de un repositorio permitirá a más sitios ofrecer y preservar sus datos.
- El uso de DSpace ha sido una correcta elección, la robustez que posee y la flexibilidad que le da el ser una herramienta libre, han permitido la adaptación a las necesidades de la colaboración.
- El uso de metodologías ágiles mejora sustancialmente la productividad, con estas es más sencillo adaptarse a los cambios y reducir los riesgos. Enfocarse en los aspectos más importantes del proyecto es otra de las ventajas.
- La implementación de identificadores persistentes garantiza el reconocimiento a los autores, especialmente en este tipo de proyecto y en esta época en la que la cantidad de contenidos generados es cada vez mayor y el acceso a estos más fácil y rápido.

- El uso de herramientas de software libre y toda la comunidad detrás de estas, permite tener productos de gran calidad, con independencia tecnológica, con alta compatibilidad y en menor tiempo. Además de unos costos muy reducidos.

8 Recomendaciones

- Debido a que el soporte para la versión 4.x de DSpace se dará hasta enero del año 2018 es muy importante que se realice la migración del repositorio en lagoproject.uis.edu.co a una versión más reciente.
- Para el año 2018 saldrá una versión de DSpace utilizando nuevas tecnologías, con esta se darán grandes cambios en la interfaz web y el entorno de desarrollo cambiará de manera considerable. Se recomienda revisar las nuevas características y planear la migración a esta nueva versión.
- Se recomienda realizar una actualización a la infraestructura de computo del servidor lagoproject.uis.edu.co, en la actualidad cuenta con una capacidad reducida de almacenamiento y el hardware utilizado no da garantías de seguridad para los datos, se recomienda un sistema de almacenamiento NAS con al menos 30 TB de almacenamiento.
- Para hacer más agradable la experiencia de usuario se recomienda, con la asesoría de un diseñador gráfico, personalizar la interfaz web.

Bibliografía

Adrian Holovaty, J. K.-M. (2009). *El libro de Django 1.0*. Apress.

Asorey, H., Cazar-Ramírez, D., Mayo-García, R., Núñez, L., Rodríguez-Pascual, M., Torres-Niño, L., et al. (2015). Data accessibility, reproducibility and trustworthiness with lago data repository.

Asorey, H., Martínez-Méndez, A., Núñez, L. A., and Valbuena-Delgado, A. (2017). Lago distributed network of data repositories. *CoRR*, abs/1704.03885.

Barbera, R., Becker, B., Carrubba, C., Inserra, G., Villalón, S. J., Kanellopoulos, C., Koumantaros, K., Núñez, L. A., Prnjat, O., Ricceri, R., et al. (2014). Chain-reds dart challenge. *ANAIS DAS SESSÕES TEMÁTICAS E PÔSTERS*, page 166.

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. (2003). Xen and the art of virtualization. *SIGOPS Oper. Syst. Rev.*, 37(5):164–177.

Collaboration, P. A. et al. (2015). The pierre auger cosmic ray observatory. *arXiv preprint arXiv:1502.01323*.

Fall, K. R. and Stevens, W. R. (2011). *TCP/IP illustrated, volume 1: The protocols*. addison-Wesley.

Forcier, J., Bissex, P., and Chun, W. J. (2008). *Python web development with Django*. Addison-Wesley Professional.

Hakala, J. et al. (2010). Persistent identifiers—an overview. *KIM Technology Watch Report*.

Heck, D., Schatz, G., Knapp, J., Thouw, T., and Capdevielle, J. (1998). Corsika: A monte carlo code to simulate extensive air showers. Technical report.

Johnston, L. R., Carlson, J., Hudson-Vitale, C., Imker, H., Kozlowski, W., Stewart, C., and Hswe, P. (2016). [preprint] data repository and curation services, how do we compare? a snapshot of six academic library institutions.

Keller, M. S. (1999). Take command: Cron: Job scheduler. *Linux J.*, 1999(65es).

Lagoze, C., Van de Sompel, H., Nelson, M., and Warner, S. (2015). The open archives initiative protocol for metadata harvesting. Technical report.

Lewis, S., de Castro, P., and Jones, R. (2012). Sword: Facilitating deposit scenarios. *D-Lib Magazine*, 18(1):4.

Luis Alejandro Torres Niño ; Directores Luis A. Núñez, C. J. B. (2016). Tratamiento y resolución de las descripciones definidas y su aplicación en sistemas de extracción de información.

Lynch, C. A. (2003). Institutional repositories: essential infrastructure for scholarship in the digital age. *portal: Libraries and the Academy*, 3(2):327–336.

Schwaber, K. and Sutherland, J. (2011). The scrum guide. *Scrum Alliance*.

Sharma, J. and Sarin, A. (2016). Getting started with spring framework: a hands-on guide to begin developing applications using spring framework.

Sidelnik, I., Collaboration, L., et al. (2015). The sites of the latin american giant observatory. In *Proceedings, 34rd International Cosmic Ray Conference (ICRC2015)*.

Team, T. D. D. (2016). Dspace 6.x documentation. Technical report.

Torres, L. A., Nuñez, L. A., Torréns, R., and Barrios, E. (2011). Implementación de un repositorio de datos científicos usando dspace. *e-colabora, "Revista de ciencia, educación, innovación y cultura apoyadas por redes de tecnología avanzada"*., 1(2):101–117.

virktualenv Developer Team, T. (2016). Virtualenv documentaton. Technical report.