



ESCUELA POLITÉCNICA



UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica
Grado en Ingeniería Informática
en Ingeniería de Computadores

Trabajo Fin de Grado

Repositorio Digital *Nopoki*

Sergio Zabala Mesonero
Noviembre, 2018



UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica
Grado en Ingeniería Informática
en Ingeniería de Computadores

Trabajo Fin de Grado

Repositorio Digital *Nopoki*

Autor: Sergio Zabala Mesonero

Tutor: Juan Antonio Gómez Pulido

Tribunal Calificador

Presidente: Julio Ballesteros Rubio

Secretario: Francisco Andrés Hernández

Vocal: Miguel Ángel Vega Rodríguez

Resumen

Este trabajo nace de una propuesta realizada por IEEE Sight España (*Institute of Electrical and Electronics Engineers, Special Interest Group on Humanitarian Technology*), junto con la *Oficina de Cooperación al Desarrollo y Voluntariado* de la *Universidad de Extremadura* (OCUDV), en el marco de una cooperación internacional con IEEE Sight Perú.

El proyecto consiste en el desarrollo de una plataforma web, concretamente un repositorio, que será utilizado para almacenar documentación en múltiples formatos (archivos digitales PDF, textos escaneados, imágenes, etc.) proveniente de comunidades amazónicas peruanas.

Aparte de almacenar dicha documentación, el repositorio de documentos también servirá para indexar su contenido, ordenarlos por categorías, asignarles etiquetas por temas, autores y otros criterios, facilitando de esta forma su búsqueda y su acceso en ámbitos académicos, históricos, antropológicos, etc.

Abstract

This project was developed from a proposal of IEEE Sight Spain (Institute of Electrical and Electronics Engineers, Special Interest Group on Humanitarian Technology) in collaboration with the Development Cooperation and Volunteering Office of the University of Extremadura (Spain), in the framework of an international cooperation project with IEEE Sight Peru.

The project consists in developing a web platform, specifically a digital repository, which will be used to store multi-format documentation (PDF files, scanned documents, pictures, etc.) from Peruvian Amazonic communities.

Besides to save that documentation, the platform will be able to index their content, sort by categories, attach keywords of topics, authors or other facts, in order to simplify the searches and the accesses in academical, historical, and anthropological scopes, among other interests.

ÍNDICE GENERAL DE CONTENIDOS

1	INTRODUCCIÓN.....	10
1.1	Motivaciones del Trabajo.....	10
1.2	Entidades involucradas.....	11
1.2.1	Oficina de Cooperación al Desarrollo y Voluntariado.....	11
1.2.2	Nopoki.....	11
1.2.2.1	Contexto.....	12
1.2.2.2	Origen del proyecto Nopoki.....	12
1.2.2.3	El Proyecto Nopoki en la actualidad.....	13
1.2.2.4	Oferta formativa.....	14
1.2.3	IEEE.....	14
1.2.3.1	Sight.....	14
2	OBJETIVOS.....	16
3	ESTADO DEL ARTE.....	17
3.1	Servidores Web.....	17
3.1.1	Historia.....	17
3.1.2	Funcionamiento.....	17
3.1.3	Servidores web más populares.....	19
3.2	Web Semántica.....	21
3.2.1	Componentes de la Web Semántica.....	22
3.2.2	Aplicaciones de la Web Semántica.....	24
3.3	Content Management System.....	25
3.3.1	Funcionalidades y aplicaciones de un CMS.....	26
3.3.2	Ejemplos de CMS.....	26
4	METODOLOGÍA.....	28
4.1	Etapas.....	28
4.2	Herramientas.....	29
4.2.1	Ordenador Portátil ASUS X550C.....	29
4.2.2	Máquina Virtual.....	30
5	ANÁLISIS.....	31
5.1	Actores.....	31
5.2	Requisitos.....	32
5.2.1	Requisitos Funcionales.....	32
5.2.2	Requisitos No Funcionales.....	33
5.3	Casos de Uso.....	34
5.3.1	Diagramas de Casos de Uso.....	34
5.3.1.1	Diagrama para el Usuario Público.....	34
5.3.1.2	Diagrama para el Usuario Publicador.....	35
5.3.1.3	Diagrama para el Usuario Administrador.....	36
5.3.2	Descripción de los Casos de Uso.....	37
5.4	Herramientas seleccionadas.....	39
5.4.1	Comparación de distintos CMS.....	39
5.4.2	El CMS elegido para el proyecto: Dspace.....	41
5.4.3	Servidor Web.....	43
5.4.4	Base de Datos.....	44

5.4.5 Sistema de Ficheros (<i>Assetstore</i>).....	44
5.4.6 Herramientas del Sistema.....	45
5.4.7 Lenguajes Utilizados.....	45
6 DISEÑO.....	46
6.1 Arquitectura.....	46
6.2 Vista.....	47
7 DESARROLLO E IMPLEMENTACIÓN.....	48
7.1 Configuración del Servidor.....	48
7.1.1 Apache Tomcat.....	48
7.1.1.1 Java.....	48
7.1.2 PostgreSQL.....	48
7.1.3 Apache Maven.....	49
7.1.4 Apache Ant.....	49
7.2 Instalación de DSpace.....	49
7.2.1 Instalación en la máquina virtual.....	49
7.2.2 Configuración de Tomcat.....	51
7.2.2.1 Webapps de DSpace.....	51
7.2.3 Prueba de funcionamiento.....	52
7.2.4 Configuración de DSpace.....	53
7.2.4.1 Agrupamiento de documentos por categorías.....	53
7.2.4.2 Sistema de etiquetado de documentos.....	54
7.2.4.3 Indexación de documentos.....	54
7.2.4.3.1 Tarea de extracción de texto.....	55
7.2.4.3.2 Ejemplo de búsqueda por texto contenido.....	55
7.2.4.4 Personalización de los formularios de subida.....	56
7.2.4.5 Configuración de los filtros de búsqueda.....	57
7.2.4.6 Configuración de la apariencia.....	58
7.2.4.6.1 Interfaz de Usuario.....	58
7.2.4.6.2 Tema.....	59
7.2.4.6.3 Internacionalización.....	60
7.2.4.6.4 Emails.....	61
7.2.4.7 Almacenamiento de Documentos.....	61
7.2.4.7.1 Assetstore.....	62
7.2.4.7.2 Almacenamiento en la nube: Protocolo WebDAV.....	62
7.2.5 Configuración del Servidor.....	64
7.2.5.1 Procesos CRON.....	64
7.2.5.2 Configuración de HTTPs.....	65
7.2.5.3 Copias de seguridad.....	66
7.2.5.3.1 Base de Datos.....	66
7.2.5.3.2 Assetstore.....	66
7.2.5.3.3 Ficheros de configuración.....	66
7.2.5.3.4 Logs.....	67
7.2.5.4 Monitorización.....	67
7.2.5.5 Optimización del rendimiento del servidor.....	68
7.2.6 Gestión de Usuarios.....	69
7.2.6.1 Usuario Administrador.....	69

7.2.6.2 Usuario Publicador.....	70
8 RESULTADOS Y PRUEBAS.....	71
8.1 Pruebas realizadas.....	71
8.2 Problemas surgidos.....	71
9 CONCLUSIONES.....	73
10 TRABAJOS FUTUROS.....	75
10.1 Servicio OAI.....	75
10.2 Migración de servicios a la nube.....	75
10.3 Funcionalidad OCR.....	76
11 BIBLIOGRAFÍA.....	77
ANEXOS.....	79
ANEXO I - Guía de Usuario.....	80
AI.1 Descripción de la Aplicación.....	80
AI.2 Requisitos.....	80
AI.3 Manual de Usuario.....	80
AI.3.1 Acceso al Repositorio Digital.....	80
AI.3.2 Visualización de Contenido.....	80
AI.3.3 Búsquedas.....	81
AI.3.4 Registro de Usuario.....	83
AI.3.5 Petición de Publicación.....	85
AI.3.6 Publicación de Archivos.....	86
AI.3.7 Visualización de estadísticas.....	90
ANEXO II - Guía del Administrador.....	92
AII.1 Creación de Comunidades y Colecciones.....	92
AII.2 Gestión de Usuarios.....	95
AII.2.1 Gestión de los grupos.....	96
AII.3 Gestión de Permisos.....	98
AII.3.1 Asignación de privilegios.....	98
AII.4 Otras funcionalidades.....	101
ANEXO III - Guía del Desarrollador.....	102
AIII.1 Requisitos.....	102
AIII.2 Instalación de la plataforma.....	103
AIII.2.1 Instalación de Apache Tomcat y sus dependencias.....	103
AIII.2.2 Instalación de Dspace y sus dependencias.....	105
AIII.3 Migración del Repositorio.....	107

Índice de figuras

Figura 1.Instalaciones de Nopoki en Atalaya.....	11
Figura 2.Monseñor Gerardo Zerdín durante una graduación.....	12
Figura 3.Estudiante nomatsigenga y estudiantes asháninkas en NOPOKI.....	13
Figura 4.Servidor web con páginas estáticas.....	18
Figura 5.Servidor web con páginas dinámicas.....	19
Figura 6.Esquema de grafo RDF.....	23
Figura 7.Grafo RDF.....	23
Figura 8.API de Freebase.....	24
Figura 9.Ejemplo de estructura de un CMS.....	25
Figura 10.Diagrama de Casos de Uso (1).....	34
Figura 11.Diagrama de Casos de Uso (2).....	35
Figura 12.Diagrama de Casos de Uso (3).....	36
Figura 13.Repositorio con Omeka.....	39
Figura 14.Ejemplo de página con Joomla.....	40
Figura 15. Repositorios Institucionales con DSpace.....	41
Figura 16.Comunidades y Colecciones (Univ. Sevilla).....	41
Figura 17.Campos definidos por Dublin Core.....	42
Figura 18.Arquitectura de DSpace.....	43
Figura 19.Arquitectura del Proyecto.....	46
Figura 20.Prototipo de Interfaz.....	47
Figura 21.Interfaz por defecto de DSpace.....	52
Figura 22.Agrupamiento por tipo.....	53
Figura 23.Etiquetas.....	54
Figura 24.Etiquetas (2).....	54
Figura 25.Texto contenido en documento.....	55
Figura 26.Resultado de búsqueda.....	56
Figura 27.Filtros avanzados.....	58
Figura 28.Tema Mirage 2 en distintos dispositivos.....	59
Figura 29.Repositorio Nopoki en distintos dispositivos.....	60
Figura 30.Email de confirmación de publicación.....	61
Figura 31.Estructura del assetstore.....	62
Figura 32.Servicio web para gestionar almacenamiento en la nube.....	63
Figura 33.Ejemplo de tarea CRON.....	64
Figura 34.Certificado SSL habilitado.....	66
Figura 35.Configuración de monitorización de Tomcat.....	68
Figura 36.Servicio web de Monit.....	68
Figura 37.Panel "Listar".....	81
Figura 38.Buscador.....	82
Figura 39.Búsqueda avanzada.....	83
Figura 40.Filtros de búsqueda.....	83
Figura 41.Resultado de búsqueda con filtros.....	83
Figura 42.Acceso a usuarios.....	84

Figura 43.Registro de usuario (1).....	84
Figura 44.Registro de usuario (2).....	84
Figura 45.Registro de usuario (3).....	85
Figura 46.Registro de usuario (4).....	85
Figura 47.Registro de usuario (5).....	86
Figura 48.Formulario de petición.....	86
Figura 49.Correo de confirmación del Administrador.....	87
Figura 50.Seleccionar colección.....	87
Figura 51.Publicar documento (1).....	87
Figura 52.Publicar documento (2).....	88
Figura 53.Publicar documento (3).....	88
Figura 54.Publicar documento (4).....	88
Figura 55.Publicar documento (5).....	89
Figura 56.Publicar documento (6).....	90
Figura 57.Publicar documento (7).....	90
Figura 58.Publicar documento (8).....	91
Figura 59.Publicar documento (9).....	91
Figura 60.Ver estadísticas (1).....	91
Figura 61.Ver estadísticas (2).....	92
Figura 62.Crear comunidad (1).....	93
Figura 63.Crear Comunidad (2).....	94
Figura 64.Crear comunidad (3).....	94
Figura 65.Crear colección (1).....	95
Figura 66.Crear colección (2).....	95
Figura 67.Crear colección (3).....	96
Figura 68.Administrar usuarios.....	96
Figura 69.Crear usuario.....	96
Figura 70.Eliminar usuario.....	97
Figura 71.Administrar grupos.....	97
Figura 72.Crear grupo (1).....	98
Figura 73.Crear grupo (2).....	98
Figura 74.Eliminar grupo (2).....	99
Figura 75.Gestionar permisos.....	100
Figura 76.Acceso a colección.....	100
Figura 77.Añadir privilegio (1).....	100
Figura 78.Añadir privilegio (3).....	100
Figura 79.Añadir privilegio (4).....	101
Figura 80.Correo de petición de permiso de publicación.....	101

Índice de tablas

ComparativaServidoresWeb.....	20
Actores.....	31

1 INTRODUCCIÓN

Siendo conscientes del importante riesgo de desaparición en el que se encuentran numerosas culturas en distintas zonas amazónicas del Perú, se ha planteado una herramienta que pueda ayudar en la preservación y divulgación de dichas culturas y sus lenguas, facilitando el acceso a una gran cantidad de documentos, que podrán ser utilizados desde cualquier parte del mundo de una forma sencilla, ya que en la actualidad no existe ningún recurso similar.

1.1 Motivaciones del Trabajo

En este apartado se enumeran los principales razones, tanto sociales como académicas, que han impulsado a realizar este proyecto:

- Ayudar a la conservación y divulgación de culturas en riesgo de desaparición o marginación, debido al escaso apoyo gubernamental con el que cuentan; en este caso, las culturas amazónicas del Perú.
- Aprovechar los recursos y medios, así como conocimientos de que disponen las universidades españolas, y más concretamente la *Universidad de Extremadura*, para lograr, en la medida de lo posible, que colectivos de entornos geográficos y sociales menos favorecidos puedan beneficiarse de dichos medios y alcanzar una mejor formación y desarrollo en dichos entornos.
- Trabajar en el ámbito de la cooperación internacional, con el enriquecimiento cultural que supone, así como la creación de relaciones entre universidades de distintos países, para la posible colaboración en futuros proyectos.
- Colaborar con una entidad de la relevancia de IEEE (*Institute of Electrical and Electronics Engineers*), lo cual otorga al proyecto un reconocimiento de carácter profesional internacional.
- Ofrecer un servicio que pueda ser utilizado más allá del contexto puramente académico de un Trabajo de Fin de Grado, ya que posteriormente será utilizado por el público en general, si bien está especialmente dirigido a expertos en materia de Historia, Antropología, Literatura, y otros muchos ámbitos que puedan consultar o aportar documentación.

- Adquirir conocimientos sobre la gestión de un servicio Web, así como de las distintas partes que lo componen (servidor, base de datos, interfaz web, etc).

1.2 Entidades involucradas

En este subapartado se presentan las distintas entidades que forman parte de este proyecto.

1.2.1 Oficina de Cooperación al Desarrollo y Voluntariado



Logo 1: Oficina de Cooperación al Desarrollo

Es un órgano dependiente del Vicerrectorado de Infraestructura y Servicios Universitarios de la Universidad de Extremadura (España) y fue creado en 2001.

Su misión es la de llevar a cabo actividades y participar en programas basados en valores como la justicia, la solidaridad, el desarrollo o el intercambio cultural, entre otros, y de los que puede tomar parte el colectivo universitario de Extremadura.

Un ejemplo de dicha participación en ese tipo de programas es el presente proyecto.

1.2.2 Nopoki

Nopoki (su nombre significa “estoy aquí” en la lengua *Shipiba*) es una Universidad sita en Atalaya, Ucayali, en el centro-este de Perú, en plena selva amazónica peruana. Se trata de un centro bilingüe al que acuden cientos de jóvenes de distintas zonas amazónicas y pertenecientes a diversas etnias nativas. [1] [2]



Figura 1. Instalaciones de Nopoki en Atalaya

1.2.2.1 Contexto

El proyecto Nopoki se desarrolla en el marco de una zona formada por poblaciones indígenas dispersas y aisladas entre sí, cada una con una lengua y una identidad propia.

Dichas comunidades subsisten a base de extraer y tratar ellos mismos los recursos de la naturaleza. Es, por tanto, un ambiente de pobreza y analfabetismo en el que los jóvenes no cuentan con opciones de obtener unos conocimientos adecuados que les permitan desarrollar esas comunidades.

El Estado solamente se preocupa de estas zonas para conceder licencias que permitan la explotación y deforestación de sus recursos.

En los últimos años, gracias a un plan gubernamental de descentralización para diseminar la actividad económica y social del país desde la capital, Lima, al resto del territorio, la ciudad de Atalaya ha podido multiplicar su población hasta alcanzar unos 40.000 habitantes, de modo que esto permite la mejora de infraestructuras y de capital humano para que pueda seguir creciendo la Universidad Nopoki.

1.2.2.2 Origen del proyecto Nopoki

Nopoki nace de la mano del religioso de origen croata, Monseñor Gerardo Zerdín, el cual recorrió la selva durante años estableciendo contacto con las distintas etnias que allí conviven (*Asheninkas, Matshiguengas, Yaneshas...*), quien tenía la ambición de formar a esos habitantes, conservando su identidad cultural y su lengua.



Figura 2. Monseñor Gerardo Zerdín durante una graduación.

Durante épocas pasadas, en los colegios locales, los profesores que impartían clases eran personas desmotivadas que habían sido delegados allí por el gobierno contra su

voluntad. Estaban, por tanto, alejados de las lenguas y las culturas locales, lo cual suponía una barrera de comprensión que derivaba en fracaso escolar.

Se incurría por tanto en una situación problemática, ya que se contaba con numerosos recursos naturales pero no se disponía de formación apropiada para poder gestionarlos de una forma eficiente.

1.2.2.3 El Proyecto Nopoki en la actualidad

El objetivo de esta institución es formar a los jóvenes en una educación bilingüe; enseñando castellano como lengua vehicular y utilizando la lengua propia de cada etnia.

Se podría haber enseñado inglés en lugar de las lenguas locales para conseguir una mayor globalización, pero esto no es lo que perseguía Monseñor Zerdín al crear Nopoki, ya que también se persigue la preservación de la identidad cultural.

Durante los cinco años de estancia de los alumnos en la Universidad, que es lo que duran sus estudios, los jóvenes son también los encargados del mantenimiento de las instalaciones, teniendo que cultivar alimentos, hacer labores de limpieza, etc.

La formación que reciben es teórica y práctica, aprendiendo a desarrollar labores de su oficio.

La convivencia entre personas de distintas etnias ha propiciado que cada vez más se relacionen entre ellos y se favorezca un sentimiento de afinidad entre los miembros de diferentes comunidades.



Figura 3. Estudiante nomatsigenga y estudiantes asháninkas en NOPOKI.

1.2.2.4 Oferta formativa

La variedad de estudios que se ofertan no es muy amplia, debido a los medios limitados de que se disponen y de el reducido número de alumnos, que rondará en torno a los 500.

Entre las carreras que se ofertan, encontramos *Magisterio bilingüe, Ingeniería agropecuaria, oficios, Inglés, etc.*

1.2.3 IEEE

El *Institute of Electrical and Electronical Engineers*, es la organización a nivel mundial más importante en el sector de las tecnologías de la información y las comunicaciones.



Logo 2: IEEE

Está formada por miles de ingenieros de diversos ámbitos (ingenieros electrónicos, en telecomunicaciones, de computadores...) así como otros profesionales como físicos o matemáticos, de los cuales una buena parte son estudiantes.

Esta asociación es la encargada de fijar estándares, con el fin de establecer unas normas comunes para un determinado área (conexiones, funcionamiento de una tecnología). Algunas de las normas más populares pueden ser los estándares *IEEE 802.X*, relacionados con las conexiones a Internet LAN y WiFi.

IEEE se divide en varias sociedades, en concreto 37, encargada cada una de un determinado área. Algunas de estas sociedades son: Comunicaciones, Sistemas y Circuitos, Educación, Robótica y Automatismos etc.

1.2.3.1 Sight

Dentro de IEEE se crea el programa SIGHT (*Special Interest Group on Humanitarian Technology*), englobado dentro del Comité de Actividades Humanitarias de IEEE (*HAC*). Este programa está formado tanto por miembros del IEEE voluntarios como personas externas, con conocimientos técnicos e interesados en trabajar en una gran comunidad internacional de actividades con gran impacto humanitario a nivel local [3].



Logo 3 : IEEE Sight

Así mismo, dichos miembros pueden realizar propuestas a la organización con proyectos que demuestren explícitamente su ayuda para una comunidad local, y que requieran ciertos requisitos tecnológicos (áreas como Agricultura, Educación, Energía, Sanidad...). De esta forma, *IEEE Sight* podrá evaluar la propuesta y conceder un soporte económico de entre 1.000 y 20.000 dólares.

A partir de este proyecto, se está elaborando una propuesta más amplia que involucra a tres universidades y dos grupos SIGHT, en la que se solicitará un soporte económico detallado y convenientemente justificado.

Algunos proyectos llevados a cabo dentro de este programa recientemente son:

- Instalación de una planta fotovoltaica para abastecer a una población de 400 habitantes en El Salvador.
- Red Inalámbrica contra catástrofes sísmicas en Nicaragua.
- Centro Educativo para aborígenes en Canadá.
- **Repositorio Digital Nopoki.**

2 OBJETIVOS

Este trabajo surge de la necesidad de crear una plataforma que pueda dar soporte de una forma persistente, accesible, simple e intuitiva, a una amplia cantidad de documentación de índole académica y cultural que será aportada por distintos expertos en materias como Historia o Antropología.

Los requisitos especificados por parte del IEEE Sight España, en cuanto a las funcionalidades que se deben cumplir, son las siguientes:

- Repositorio que permita almacenar ficheros PDF y otros contenidos multimedia.
- Funcionalidad de anotación semántica para documentos textuales y archivos digitales multimedia.
- Funcionalidad de indexación de términos y etiquetas de los documentos textuales y multimedia.
- Motor de búsqueda semántico y por contenido.
- Funcionalidad de similitud de documentos textuales y multimedia en base a contenido y etiquetas.

Por parte del desarrollador del proyecto, se han fijado otros objetivos que pueden servir para proporcionar mayor robustez y versatilidad, como son:

- Utilizar herramientas Software de código abierto (*open source*), lo cual está en mayor consonancia con el carácter cooperativo y voluntario del proyecto.
- Garantizar la *disponibilidad* del servicio, por medio de herramientas que controlen el estado de los recursos, y actúen en consecuencia para reestablecer el servicio automáticamente o notifiquen los problemas al administrador, a fin de solucionarlos a la mayor brevedad posible.
- Depositar los documentos del repositorio, así como las configuraciones y otros datos de interés, en un medio que ofrezca disponibilidad y seguridad (en este caso, la nube), para poder acceder a ellos de forma continua, y que esa información no se pierda aunque se produzca un fallo en el servidor, en cuyo caso se pueda hacer una recuperación.
- Aumentar la *seguridad* por medio del uso de protocolos seguros para el acceso, así como limitar el acceso a ciertos recursos.

3 ESTADO DEL ARTE

3.1 Servidores Web

Un servidor web es un programa situado en el equipo que actúa como servidor (un servidor dedicado, una máquina virtual, un ordenador convencional u otro tipo de soporte) y que efectúa conexiones con la aplicación del equipo cliente, recibiendo peticiones por parte de este y entregando datos almacenados que le serán presentados como un sitio Web. [4]

3.1.1 Historia

Los servicios web se crean para conseguir una comunicación universal entre distintas plataformas y distintos lenguajes de programación a través de Internet.

Fue a partir de finales de los años 80, cuando el científico Tim Berners-Lee; creador de la World Wide Web, desarrolló una computadora denominada *NeXTCube*; la cual se utilizaría por primera vez como servidor web.

El propio Berners-Lee diseñó un sistema de intercambio de información basado en hipertexto, que posteriormente junto a Robert Cailliau crearían un navegador web y los primeros elementos que componen Internet; el Lenguaje HTML (*HiperText Markup Language*) y el protocolo HTTP (*HiperText Transfer Protocol*). [5]

Ya en la década de los 90 se comenzaron a conectar equipos de forma global, comunicándose los PCs con los servidores independientemente del Sistema Operativo, el lenguaje de programación y se diseñan los servicios web basados en XML (*Extensible Markup Language*), para que las aplicaciones puedan compartir su información entre ellas tanto dentro como fuera de la empresa.

3.1.2 Funcionamiento

Un servicio web sigue el modelo Cliente/Servidor [6] [7]. Aplicado al caso de esta aplicación web, las fases que se siguen para que un usuario pueda utilizar dicha aplicación explicadas de una forma general son:

1. El usuario (Equipo Cliente) accede a la página web del repositorio. El navegador web del usuario establece una conexión TCP mediante el protocolo HTTP con el equipo servidor para obtener la página.

2. Una vez realizada esta primera conexión , el Servidor realizará una serie de acciones antes de responder la solicitud del cliente (comprobar si existen los recursos solicitados , si el usuario tiene permisos para acceder a ellos,etc).

El funcionamiento del equipo servidor a la hora de atender a las peticiones puede variar en función de si la página es *estática* (con recursos que no varían como páginas HTML, imágenes, textos, etc) o *dinámica* (se ejecutan programas en el equipo de la parte del servidor):

- 2.1 Si es *estática*: Se establecerá una conexión HTTP por cada elemento que componga la página, el servidor enviará dichos elementos a la parte del cliente para que éste pueda ver dicha información en el navegador.

Se puede representar este proceso con un esquema de este tipo:

1 – Cliente solicita una página web mediante un navegador, por ejemplo.

2 – La aplicación del servidor web, alojada en el Equipo Servidor, atiende a esa petición y se encarga de localizar los recursos.

3 – El Servidor devuelve la página con los recursos, que se muestra en el navegador.

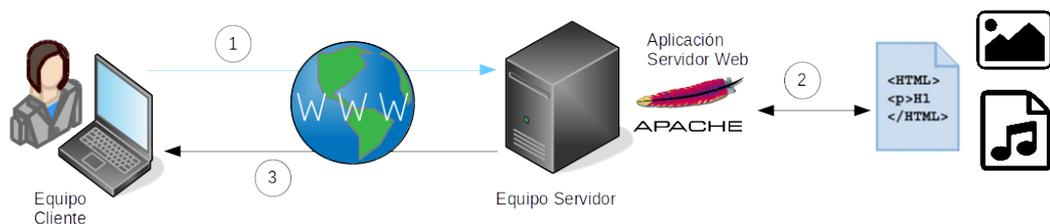


Figura 4. Servidor web con páginas estáticas

- 2.2 Si es *dinámica*: Al igual que en el caso de las páginas estáticas, el cliente realiza una conexión HTTP por cada elemento de la web.

En la parte del equipo servidor, además, en la propia página HTML existen fragmentos de código denominados *script*, escritos en lenguajes como PHP, JSP o Python, por ejemplo, que se ejecutan en el servidor y cuyo resultado se devuelve integrado en la propia página HTML que se devuelve al cliente.

Durante la ejecución de dicho código o *script*, el servidor puede acceder a recursos como Bases de Datos , sistemas de ficheros, otros servidores ,etc.

El esquema representativo difiere del anterior, siendo de esta forma:

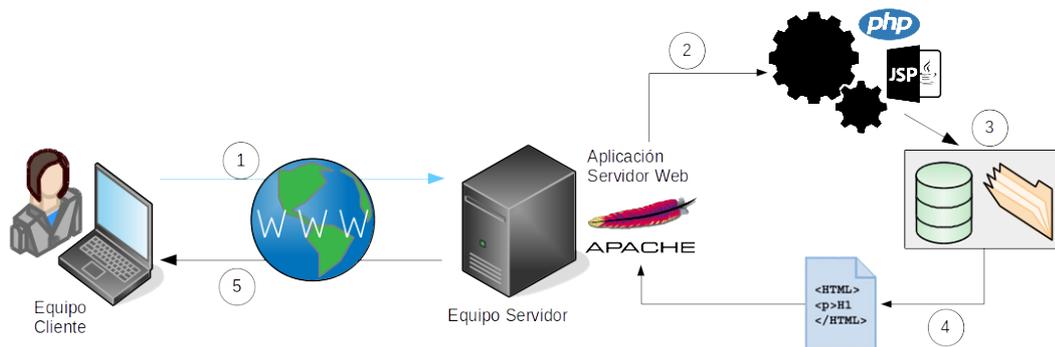


Figura 5. Servidor web con páginas dinámicas

1 – Cliente solicita una página web mediante un navegador, por ejemplo.

2 – La aplicación del servidor web, alojada en el Equipo Servidor, atiende a esa petición y ejecuta los scripts (código en lenguajes PHP, JSP, Python...) integrados en la página.

3 – Durante la ejecución de dicho código, el servidor puede acceder a otros recursos, como Bases de Datos, Sistemas de ficheros, Servidores de correo, etc.

4 – El resultado de la ejecución de los scripts se inserta en la página HTML que será entregada al cliente.

5 – El Servidor devuelve la página al Equipo Cliente, y será mostrada en su navegador web.

3.1.3 Servidores web más populares

En la actualidad, existe cierta variedad en el software dedicado a servidores web, y la elección de uno u otro por parte del administrador dependerá de la naturaleza de la página, el código que esta incluya, si será utilizado como servidor frontal (al que se accede desde Internet) o como servidor proxy (servidor intermedio entre el cliente y el servidor frontal), entre otras variables a tener en cuenta.

Aquí se realiza una comparativa de los más utilizados a día de hoy, explicando sus principales características:

Nombre	Descripción	Licencia	SO
Apache HTTP	Es el más popular. Se estima que representa un 70 % de los servidores web existentes. Se suele utilizar en entornos Linux, con Bases de Datos MySQL y con scripts PHP, aunque también Python o Perl (Configuración <i>LAMP</i> : <i>Linux</i> + <i>Apache</i> + <i>MySQL</i> + <i>PHP</i>). [8]	Open source	UNIX, Windows
Apache Tomcat	También creado por la <i>Apache Software Foundation (ASF)</i> . La característica de esta herramienta es que implementa <i>Servlets de Java</i> (código en Java compilado y ejecutado en el equipo servidor) y <i>JavaServer Pages (JSP)</i> , utilizadas para crear aplicaciones web. Es por esto que en el equipo del servidor deberá estar instalada alguna versión de Java. Este es el utilizado para el presente proyecto. [9] [10]	Open source	UNIX, Windows
Nginx	Aplicación de servidor que ofrece una alta concurrencia y bajo uso de memoria, debido a su gestión de las solicitudes web. Aparte de su uso como servidor web convencional, es muy frecuente su uso como servidor proxy, caché o balanceador. [11]	Open source	UNIX, Windows
Microsoft IIS	El <i>Internet Information Server</i> de <i>Microsoft</i> es un servidor multi propósito que permite ofrecer recursos en redes locales (LAN), redes corporativas (<i>Intranet</i>) o en Internet. Otra de sus características es que soporta los scripts <i>Active Server Page (ASP.NET)</i> , para las páginas dinámicas. Es el segundo más usado globalmente. [12]	Propietaria	Windows

3.2 Web Semántica

La *Web Semántica* (también conocida como *Web 3.0*) es un complemento de la web actual, que cumple el cometido de asignar significado, lo cual se conoce como semántica, a los datos contenidos en Internet.

Fue impulsada por el anteriormente mencionado creador de la *World Wide Web*, Tim Berners-Lee, debido a la necesidad de ordenar la gran cantidad de información existente en la red, ya que el lenguaje HTML sólo proporciona las normas para mostrar los datos, pero sin describir dichos datos.

El principal objetivo de esta extensión de la web es la de facilitar el acceso al usuario a dicha información, de modo que cuando realice una búsqueda, por ejemplo, no sólo se le enumeren los resultados que contengan las palabras clave sin ningún orden, si no que reciba en dichos resultados el tipo de recurso que más se amolde a lo que el usuario estaba solicitando. [13] [14]

Esto se consigue, por poner un ejemplo que clarifique el concepto de web semántica; agrupando la información en *entidades*. Dichas entidades representan elementos existentes en la vida real, y contendrán un gran número de palabras y conceptos clave que las definan. [15]

Un ejemplo de dichas entidades podría ser: “Universidad de Extremadura”. En dicha entidad se podrían englobar palabras como:

- “Universidad de Extremadura”
- “UEX”
- “Unex”
- “Univ. de Extremadura”
- “Universidad de Cáceres”
- “Universidad del oeste de España”
- ...

De este modo, cuando un usuario busque alguno de esos conceptos, le será mostrada información relacionada con la Universidad de Extremadura, en lugar de mostrar páginas que contengan esas palabras clave sin ningún orden lógico.

Las entidades están relacionadas entre sí, de modo que es posible buscar elementos que compartan características comunes.

De esta forma, la web semántica tiene la capacidad de que las aplicaciones puedan intercambiar información entre sí, ya que todas las entidades son accesibles por parte de la comunidad online.

Esto proporciona la ventaja, por ejemplo, de reutilizar un trabajo por parte de otras entidades, evitando así aportar información redundante que ya existe.

3.2.1 Componentes de la Web Semántica

Las principales partes que componen la web semántica son:

- **XML(*eXtensible Markup Language*):** El XML (o Lenguaje de Marcas Extensible) es un lenguaje desarrollado por W3C para estructurar los datos contenidos en una web, otorgando un significado a los datos sin ninguna restricción, agrupándolos en elementos, delimitados por etiquetas
 - *XML Schema*: Plantilla que define la estructura de los documentos XML.
- **URI (*Uniform Resource Identifier*):** Cadenas de texto que identifican de manera universal todos los recursos publicados en Internet. Pueden ser :
 - *Nombre Uniforme de Recurso (URN)*; que no indica exactamente dónde está el recurso.
 - *Localizador Uniforme de Recursos (URL)*; que es una dirección que hace referencia a un elemento de Internet, aunque este puede ser diferente a lo largo del tiempo; es decir, puede hacer referencia a varios.
 - La suma de ambos (URN + URL)

El URI consta de las siguientes partes:

- Esquema: Indica el protocolo con el que se accede (*http, ldap, ftp*).
- Máquina: Nombre del sitio o *host* donde se aloja el recurso (p.ej: www.sitio.net)
- Otros campos: Directorio , Puerto , Usuario y Contraseña...

Como ejemplos de URI completos: <https://sight.unex.es:8080/solr> ,

<jdbc:postgresql://myserver:5432/dspace>

- **RDF (*Resource Description Framework*):** La Infraestructura de Descripción de Recursos es un lenguaje simple para representar los metadatos (título, autor, copyright ,etc) de los recursos (imágenes, directorios) que componen la web. Es por esto que se le define como un modelo de datos. Es

muy útil para el intercambio de información entre aplicaciones, como buscadores, más que para la lectura por parte de humanos. [16]

Para la identificación de los distintos componentes de una web, RDF utiliza los URI, y para describirlos mediante propiedades simples y valores utiliza las *sentencias o declaraciones*, que son tripletas (conjuntos de 3 valores) formados por los campos: [17]

- *Sujeto*: Recurso que se describe, identificado por un *URI* (p.ej: una página web)
- *Predicado*: Propiedad de dicho recurso (*autor, idioma, fecha ...*)
- *Objeto*: Valor de la propiedad (“Nombre del autor”, “Español”, “20-7-2016”, ...)

Un ejemplo de los valores de una declaración para un recurso:

- *Sujeto*: Página web identificada con el URI: <http://www.miweb.com>
- *Predicado*: Fecha de creación, identificada con el URI: <http://www.miweb.com/terms/creation-date>
- *Objeto*: Valor de la propiedad “Fecha de creación”: “20/02/2014”

Otra manera de representar esta sentencia, es mediante un grafo del tipo:



Figura 6. Esquema de grafo RDF

Para el ejemplo expuesto, el grafo resultante sería:

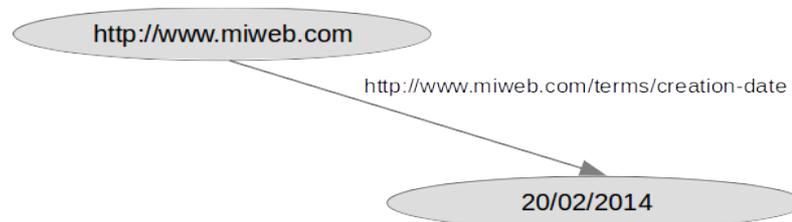


Figura 7. Grafo RDF

- **OWL (*Ontology Web Language*):** El lenguaje ontológico se encarga de establecer relaciones entre recursos, agrupándolos en áreas de conocimiento.

3.2.2 Aplicaciones de la Web Semántica

Aquí se enumeran algunos ejemplos de posibles usos de la Web Semántica:

1. **Redes Sociales (Facebook, Twitter, Flickr...):** Permite asignar datos informativos (fecha, autor, lugar, etc) a las imágenes subidas, por ejemplo, para facilitar la conexión de otros miembros.
2. **Búsqueda de Bibliografía:** Consultando en colecciones digitales de bibliografía que están relacionadas.

Así mismo, estos son casos reales de compañías o herramientas que se benefician de la Web Semántica: [18]



Logo 4.
RSS

1. **RSS (*Really Simple Indication*):** Tecnología usada para recibir contenido de una página web de una forma sencilla sin tener que utilizar un navegador, ya que basta con un software capaz de leer la información a la que se suscriba, por ejemplo: Diarios digitales, redes sociales etc.
2. **Open Graph Protocol:** Tecnología utilizada por la red social Facebook para que se pueda interactuar con contenido de esta sin necesidad de acceder, por ejemplo: Un usuario puede compartir un artículo en su cuenta personal de Facebook desde otra web.
3. **Freebase:** Plataforma colaborativa en la que la comunidad puede aportar datos de forma estructurada que se almacenan en una base de datos.

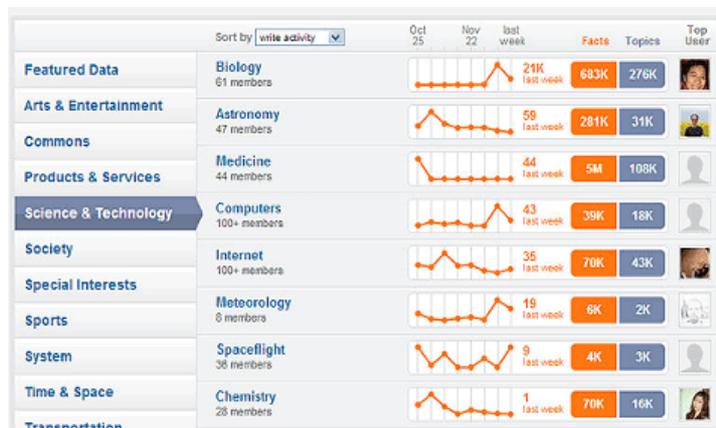


Figura 8.API de Freebase

3.3 Content Management System

Un *Content Management System* (abreviado con la siglas *CMS*) o *Sistema Gestor de Contenidos*, es un programa informático utilizado para organizar los recursos existentes en una web. [19]

Esta herramienta permite crear, editar y publicar contenidos en la web a múltiples contribuidores mediante una interfaz intuitiva, evitando así que el creador deba escribir todo el código para implementar las funcionalidades de la página, permitiendo a usuarios con menos conocimientos técnicos de desarrollo crear nuevos sitios web, así como su administración.

En cuanto a su arquitectura; por norma general se estructuran en tres capas:

- **Capa de Datos:** En la que se almacenan toda la información contenida en la web, por lo suele disponer de una Base de Datos.
- **Capa de Aplicación:** Encargada de recoger los datos almacenados y mostrárselos al usuario.
- **Capa de Presentación:** En la que se define la apariencia visual que será mostrada a través del navegador.

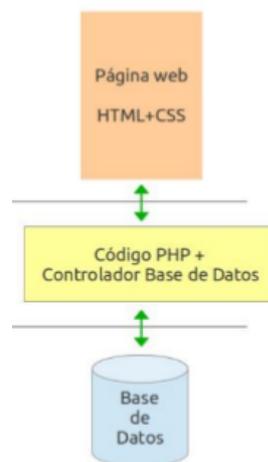


Figura 9. Ejemplo de estructura de un CMS

3.3.1 Funcionalidades y aplicaciones de un CMS

Algunas de las soluciones principales que proporciona un CMS son: [20]

- **Creación de sitios web:** Permiten crear nuevas páginas con conocimientos básicos de programación, personalizando la apariencia visual de una forma sencilla sin afectar al contenido.
- **Administración de webs:** Facilita la administración de entornos webs debido a que el CMS se encarga de gestionar las Bases de Datos, y cuenta con herramientas como extensiones (o *addons*) para añadir funcionalidades, actualizar contenidos etc.
- **Búsqueda de contenidos:** Facilita el acceso a los contenidos , sin necesidad de conocer dónde están almacenados, gracias a tecnologías como los *metadatos* o la *indexación*, propias de la web semántica.
- **Soporte multilingüe:** Permiten la visualización en múltiples idiomas sin necesidad de reescribir el código entero para cada uno.
- **Seguridad:** Utilización de protocolos para encriptar información, actualizaciones periódicas etc.

Entre los usos más frecuentes en los que se suelen emplear los CMS, podemos encontrar: [21]

- **Sitios webs corporativos**
- **Blogs personales**
- **Tiendas virtuales (Plataforma eCommerce)**
- **Repositorios digitales**

3.3.2 Ejemplos de CMS

Aquí se enumeran algunos ejemplos reales de CMS utilizados actualmente y algunas de sus características:

- **WordPress:** Es el Gestor de Contenidos más popular y extendido, debido a su sencillez y su flexibilidad , ya que existen numerosas plantillas para modificar su apariencia así como *plugins* para añadir capacidades.



Logo 5.
WordPress

- **Joomla:** CMS de código abierto para crear todo tipo de sitios web (tiendas virtuales, foros, galerías de imágenes etc) con multitud de componentes adicionales totalmente gratuitos.



Logo 6. Joomla

- **Drupal:** Similar a Joomla, se utiliza para la creación de webs multi-propósito.



Logo 7.
Drupal

- **PrestaShop:** Gestor de contenidos enfocado a la creación de plataformas de *eCommerce*, para ayudar a las empresas a vender sus productos online.



PrestaShop

Logo 8.
PrestaShop



DSpace

Logo 9. DSpace

- **DSpace:** Aplicación de código abierto enfocada a la gestión de repositorios digitales de documentación.

Facilita la búsqueda y el acceso a contenidos por medio de filtros de búsqueda, capacidad de ordenar por distintos parámetros (fecha de creación, autores etc), estructuración en comunidades, colecciones ...

Está muy extendido entre las instituciones académicas, por ejemplo. Es el seleccionado para el desarrollo del presente proyecto , por lo que se extiende información en los sucesivos apartados.

4 METODOLOGÍA

En el presente apartado se describen de una manera detallada los pasos seguidos durante el desarrollo de la solución a la propuesta planteada para el proyecto.

Se explican los planes iniciales en el diseño y cómo éstos han ido cambiando y los motivos de dichos cambios, hasta afinar una solución final que se amolde a los requisitos del sistema pero teniendo en cuenta las limitaciones de recursos, tiempo y presupuesto.

4.1 Etapas

Se podría dividir el desarrollo del proyecto a lo largo del tiempo en distintas etapas, si bien dichas etapas no son sucesivas en el tiempo, si no que en muchos casos se realizan en paralelo o se produce retroalimentación, volviendo a etapas anteriores a medida que el proyecto va creciendo.

Dichas etapas o fases podrían enumerarse como:

- **Análisis del problema:** Se han establecido reuniones telemáticas con el personal de la universidad peruana, ya que es a quién va destinado. Un ingeniero informático de la *Universidad Católica Sedes Sapientiae* de Lima (Perú) realiza la función de nexo entre la Universidad de Extremadura y la Comunidad Nopoki. Dicha persona es la encargada de fijar una serie de requisitos, así como de aceptar propuestas por parte de los desarrolladores, y podrá ir probando la aplicación a lo largo de su implementación.
 - **Selección de las herramientas adecuadas:** Estudio de diferentes soluciones software, con el requisito de que sean de código abierto, que puedan soportar las características propias de un repositorio de documentación digital. Se han comparado y probado diversas herramientas a lo largo de esta fase, especialmente los gestores CMS, para evitar tener que reescribir partes de código de la web que se puedan reutilizar. Se han ido descartando programas; como *Omeka* o *Joomla*, y se ha dado con una solución óptima que se adapta muy bien a nuestro propósito; el sistema gestor de contenidos *Dspace*, el cual se detallará en sucesivos apartados.
- **Diseño:** Se procede a elaborar un esquema de una arquitectura que cumpla los objetivos marcados en base a la herramienta seleccionada, contemplando

aspectos como la forma de mostrarlo al usuario, el almacenamiento de los datos o los distintos roles de usuario, entre otros.

- **Implementación de la solución:** Es la etapa principal y más extensa en el tiempo, ya que consiste en instalar , configurar y desarrollar el código necesario para crear el repositorio digital, así como algunos elementos auxiliares; como la seguridad o la recuperación ante fallos; y cuyos pasos se enumeran en los siguientes apartados.
- **Prueba de la aplicación:** Esta etapa se hace conjuntamente con la fase de implementación, ya que a cada funcionalidad que se añade, o cambio que se realiza es preciso probar su correcto funcionamiento y en caso contrario, hacer los ajustes convenientes o directamente revertir dichos cambios y volver al estado anterior, como ha ocurrido en diversas ocasiones.
- **Elaboración de la documentación:** A parte de la presente memoria final ,que se redacta conforme se va terminando el proyecto , se van tomando notas sobre todos los pasos realizados, así como los problemas surgidos y la solución que se ha tenido que tomar, por si vuelven a ocurrir.
- **Reuniones regulares con profesorado:** A lo largo del desarrollo del proyecto se han realizado periódicas citas con el Tutor, tanto presenciales como a través de correo electrónico, para realizar un seguimiento del mismo. En dichas reuniones, se han ido exponiendo los avances del proyecto, así como preguntado las dudas pertinentes y anotado las propuestas que el Tutor pudiese tener. También se ha buscado apoyo en más personal docente para conseguir información sobre las herramientas empleadas.

4.2 Herramientas

Se procede a enumerar y describir cada una de las herramientas implicadas como infraestructura en la realización de este trabajo, y el motivo de su uso.

4.2.1 Ordenador Portátil ASUS X550C

Equipo principal de trabajo principal, desde el cual se han realizado todas las instalaciones. Sus principales características:

- Memoria RAM: *8 GB*
- Almacenamiento en Disco: *256 GB en disco SSD + 1TB en disco HDD externo*

- Procesador: *Intel™ Core i7-3537U*
- Tarjeta gráfica: *NVIDIA GeForce GT 720 M*
- Sistema Operativo: *Linux Mint 18.1 Serena (Distribución GNU/Linux basada en Ubuntu-Debian)*

4.2.2 Máquina Virtual

Proporcionada por el Servicio de Informática de la Escuela Politécnica (*IEPCC*), en lo que se conoce como *Infraestructura como Servicio (IaaS)*.

Esta máquina será la que cumpla la función de servidor, de manera que todas las herramientas necesarias para el funcionamiento del proyecto estarán instalados en ella (CMS, Base de Datos, Servidor Web, dependencias...).

El propio servicio técnico de la EPC y sus administradores de sistemas serán los encargados de la gestión de esta máquina (realizado de copias de seguridad, instalación de paquetes con permisos específicos, gestión de los puertos etc).

En cuanto a sus características técnicas:

- Memoria RAM: 64 GB
- Almacenamiento: 200 GB
- Virtualizada con Linux Containers (*LxC*) mediante *Proxmox VE*
- Procesador del Servidor: *Intel™ Xeon E5-2650 v3*



Logo 10. Linux Containers

5 ANÁLISIS

Esta etapa del proyecto abarca desde el proceso de identificación de los requisitos a cumplir por la aplicación web, hasta la selección de las herramientas adecuadas para lograr cumplirlos

5.1 Actores

Se ha definido una serie de roles en función de los permisos que dispongan a la hora de interactuar con el sistema. En la siguiente tabla se enumeran estos roles y se detallan sus características:

Actor	Características
Usuario anónimo/público	Es la persona que accede al servicio sin registrarse en el sistema. Podrá acceder a toda la documentación para realizar búsquedas y descargar documentos. Podrá ver las estadísticas de los documentos más visitados.
Usuario Publicador	Tiene de base las mismas capacidades que el usuario anónimo. Una vez registrado, un Usuario Administrador puede darle permisos para añadir o borrar documentación de una o varias comunidades y colecciones. Podrá ver las estadísticas de los documentos más visitados.
Usuario Administrador del Repositorio	Es un usuario creado desde el servidor. Tiene permisos para gestionar todos los aspectos del repositorio. Puede crear/borrar/editar comunidades, colecciones y elementos. Puede otorgar o retirar permisos a usuarios, así como denegarles su registro como tales. Puede crear grupos que contengan varios usuarios para darles un mismo rango de permisos. Tiene acceso a estadísticas más detalladas, como las actividades realizadas en determinado período de tiempo (n.º de búsquedas, n.º de archivos subidos etc).
Administrador del sistema	Es el encargado de la instalación y gestión de toda la infraestructura sobre la que se asienta el servicio del repositorio. Instala y configura la aplicación, el servidor web, la base de datos etc. Lleva a cabo el control del estado del servicio mediante un sistema de monitorización que se comunica con su correo electrónico.

5.2 Requisitos

En base a los objetivos (*Apartado 2: Objetivos*) extraídos a partir de las sucesivas reuniones con el solicitante del proyecto, se han fijado una serie de requisitos que debe cumplir el sistema, tanto funcionales: que describen el comportamiento de la aplicación para con el usuario; como No Funcionales; que fijan especificaciones de calidad del sistema.

5.2.1 Requisitos Funcionales

- **RF1 - Almacenamiento de contenido digital:** Se permite al usuario (si está registrado en el sistema) la publicación de documentación en múltiples formatos (Texto, Audio, Gráficos etc) en un repositorio mediante un sencillo formulario. También se permite el acceso a todos los usuarios, registrados o no, a estos documentos y a su posible descarga.
- **RF2: Anotación semántica del contenido del repositorio:** El usuario que publique un documento podrá asignarle una serie de palabras clave o etiquetas para darle una semántica. Esto facilitará su posterior búsqueda y acceso.
- **RF5: Catalogación de documentos:** A parte de las etiquetas, se permitirá encuadrar cada documento en una determinada categoría (p.ej: *Artículo, Texto literario, Imagen etc*).
- **RF3: Indexación de textos:** En el caso de documentos de textos digitalizados (en formato PDF, fundamentalmente) se realizará una indexación automática del contenido de dicho documento, a fin de poder buscar fragmentos de ese texto para su localización.
- **RF5: Búsqueda por distintos parámetros:** El usuario (registrado o no) que desee realizar una búsqueda de un documento en concreto o de varios documentos similares, tendrá la posibilidad de hacerlo según diferentes ámbitos; Por categorías, por palabras clave, por autor, por país etc.
- **RF6: Organización de documentos por parámetros más relevantes:** Se facilita el listado de la documentación por los parámetros más relevantes; La fecha, el autor y la categoría.
- **RF7: Creación de Comunidades y Colecciones:** Los usuarios con permisos específicos podrán crear comunidades, divididas a su vez en colecciones, en las que posteriormente incluir nuevos documentos. De esta forma se consigue dar una estructura ordenada del repositorio.

5.2.2 Requisitos No Funcionales

- **RNF1 – Usabilidad:** Ofrecer una interfaz web sencilla e intuitiva, sin sobrecargar las opciones.
- **RNF2 – Diseño adaptativo:** Utilizar un formato *responsive*, que se adapte a múltiples soportes (ordenadores, móviles o tablets).
- **RNF3 - Disponibilidad y Recuperación ante fallos:** Disponer de un sistema de monitorización que alerte al administrador de problemas en el funcionamiento del servicio, así como establecer una política de copias de seguridad a un medio fiable como la nube o *cloud*, para almacenar la documentación, así como las copias de la Base de Datos y los ficheros de configuración esenciales.
- **RNF4 – Seguridad:** Utilización del protocolo HTTPS para el acceso a toda la web, así como la limitación de acceso a ciertos recursos más sensibles.

5.3 Casos de Uso

En el siguiente apartado se divide el comportamiento general del sistema en distintos casos de uso, cada uno describirá una funcionalidad.

Dichos casos de uso se representan gráficamente mediante diagramas, y posteriormente se detallan.

5.3.1 Diagramas de Casos de Uso

5.3.1.1 Diagrama para el Usuario Público

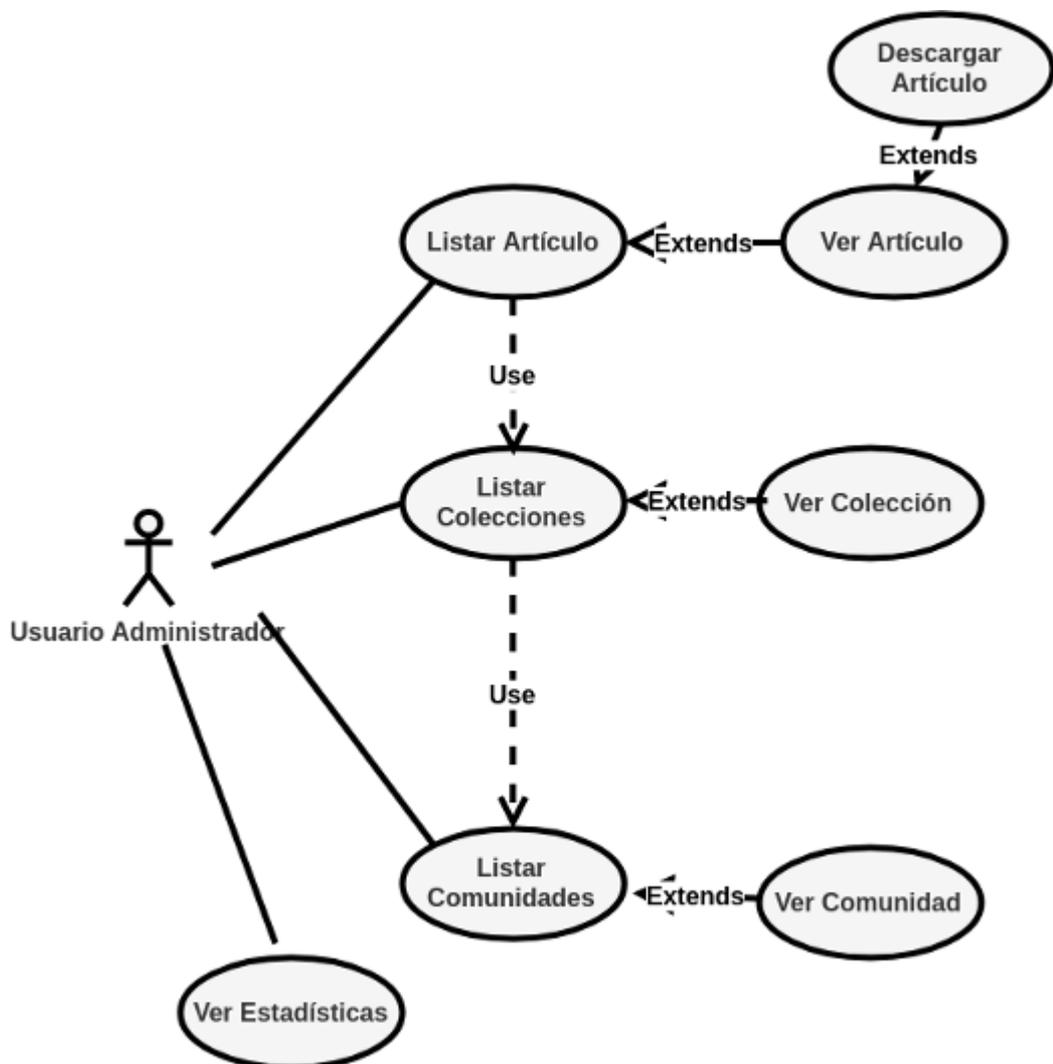


Figura 10. Diagrama de Casos de Uso (1)

5.3.1.2 Diagrama para el Usuario Publicador

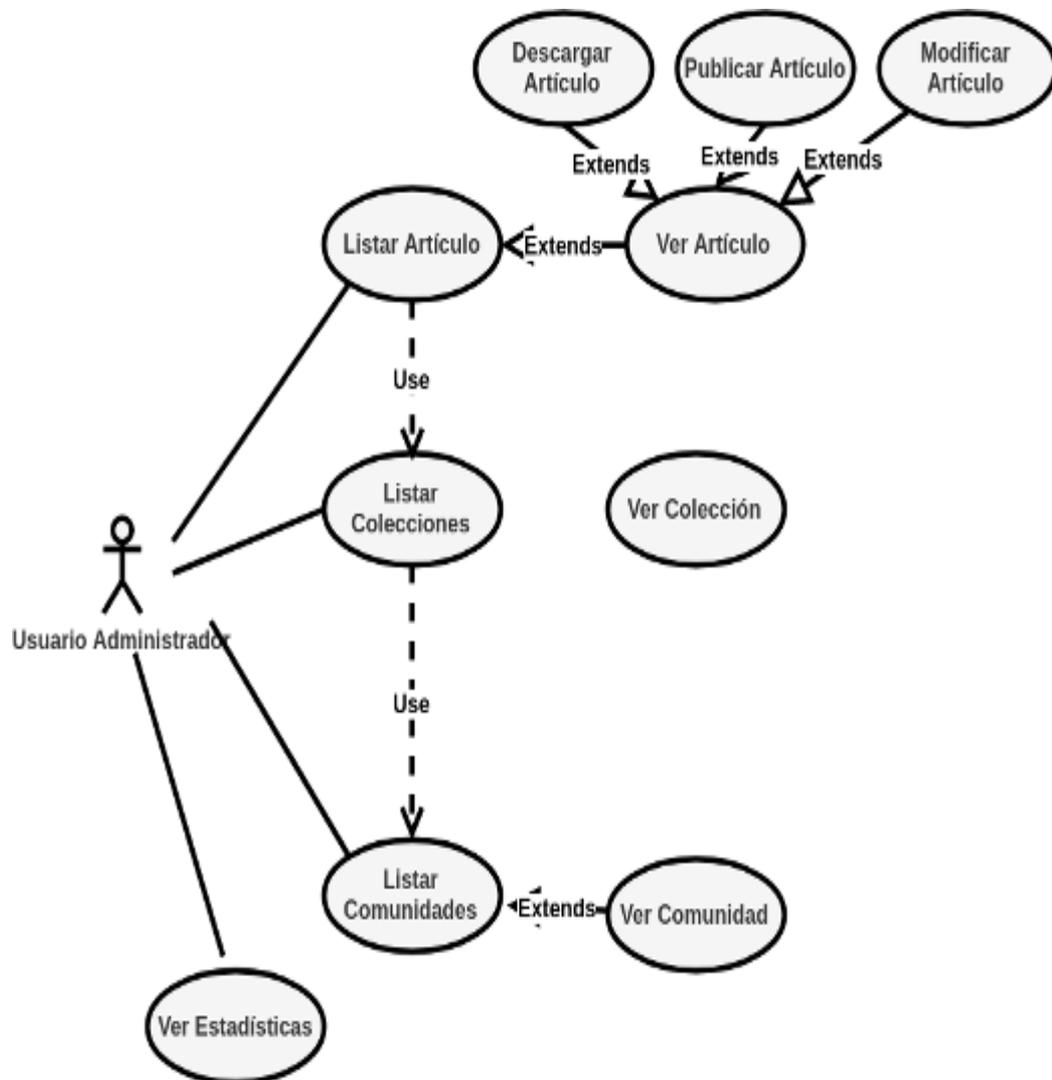


Figura 11. Diagrama de Casos de Uso (2)

5.3.1.3 Diagrama para el Usuario Administrador

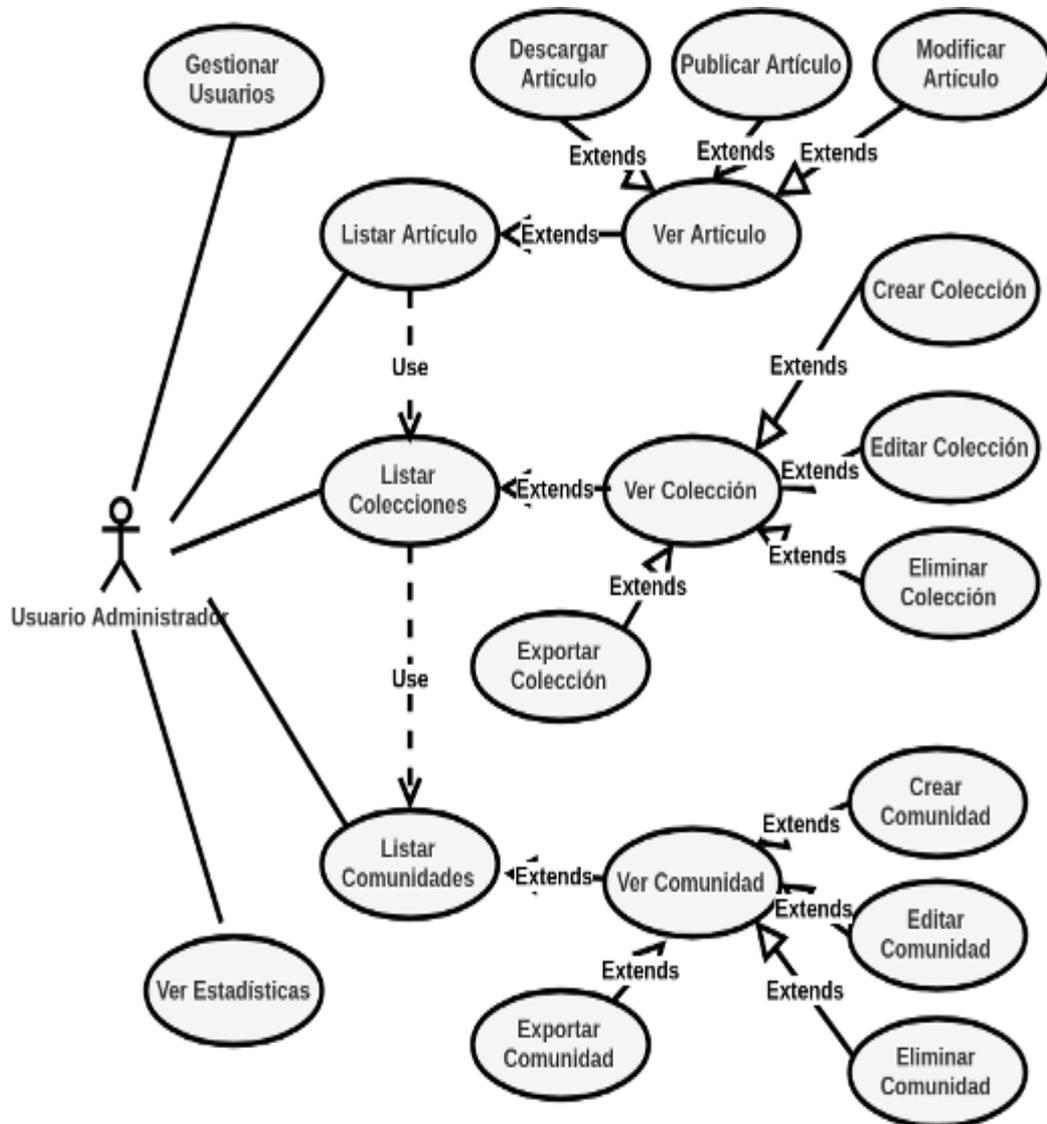


Figura 12. Diagrama de Casos de Uso (3)

5.3.2 Descripción de los Casos de Uso

A continuación se detallan las actividades realizadas en cada caso de uso para comprender en qué consisten.

- **Listar Artículos:** El usuario puede visualizar listados de artículos cuyo contenido dependerá de los parámetros que haya escogido. P.ej: Si el usuario decide listar los documentos por una determinada fecha de publicación se mostrará un conjunto de artículos que cumplan dichas características. Se asigna a todos los usuarios.
- **Ver Artículos:** El usuario puede acceder a los detalles de un determinado artículo seleccionado, de una forma simple o mostrando un registro completo con todos los metadatos que lo describen. Pueden hacerlo todos los tipos de usuario.
- **Listar Comunidades:** Visualizar todas las comunidades existentes en el repositorio. Para todos los usuarios.
- **Ver Comunidad:** Acceso a una determinada comunidad. Puede hacerlo cualquier tipo de usuario.
- **Listar Colecciones:** Cualquier usuario puede contemplar todas las colecciones de una determinada comunidad.
- **Ver colección:** Todos los usuarios pueden acceder a una colección, que contendrá una serie de artículos.
- **Descargar artículo:** Todos los usuarios pueden descargar los ficheros adjuntos a un artículo, y en el caso de usuarios con privilegios, también el registro de metadatos de este.
- **Ver estadísticas:** Todos los usuarios pueden ver las estadísticas de los documentos más usados. El usuario Administrador, además, podrá comprobar estadísticas más detalladas sobre otros ámbitos.
- **Publicar Artículo:** Los usuarios Administradores, así como los registrados que tengan los permisos pertinentes, podrán suscribir un nuevo documento al repositorio, de manera que se crea un nuevo artículo dentro de una determinada colección de una comunidad, previamente creadas.
- **Modificar Artículo:** Posibilidad del usuario de editar todos algunos aspectos del ítem; metadatos (autor, fecha, tipo etc), privilegios (qué usuarios pueden acceder), archivos adjuntos o cambiarlo de colección, entre otros. En el caso

del Administrador, se podrán gestionar los permisos sobre dicho artículo (Lectura, edición etc).

- **Crear Colección:** Añadir una nueva colección dentro de una comunidad existente (p.ej: la colección *Facultad de Derecho* dentro de una comunidad llamada *Tesis Doctorales*). Sólo para el rol Usuario Administrador.
- **Editar Colección:** El usuario Administrador puede editar los detalles de una colección creada mediante metadatos, así como gestionar los permisos de acceso y edición a esta.
- **Exportar Colección:** El usuario Administrador puede exportar los metadatos de una colección, que serán enviados directamente a su correo electrónico.
- **Eliminar Colección:** Retirar una colección del repositorio. Será exclusivo del usuario Administrador.
- **Crear Comunidad:** Añadir una nueva comunidad en el repositorio para estructurar mejor los artículos. Se asigna al rol Usuario Administrador.
- **Editar Comunidad:** El usuario Administrador puede cambiar las propiedades de una comunidad creada, así como gestionar los permisos de acceso y edición a esta.
- **Exportar Comunidad:** El usuario Administrador puede exportar los metadatos de una comunidad, que serán enviados directamente a su correo electrónico.
- **Eliminar Comunidad:** Retirar una comunidad del repositorio. Función sólo realizada por el usuario Administrador.

5.4 Herramientas seleccionadas

5.4.1 Comparación de distintos CMS

En la búsqueda de un sistema *Gestor de Contenidos* que se adecuase a nuestro propósito de almacenar grandes cantidades de documentos de índole fundamentalmente académica, científica o histórica, se han analizado algunas de las soluciones existentes de código abierto, comparándolas entre ellas y probándolas hasta llegar al sistema que más se adecua a nuestro fin.

Omeka

Es un software de código abierto enfocado a albergar colecciones de documentación digital, ideal para bibliotecas o archivos históricos.

Esta herramienta destaca por su instalación sencilla, escalabilidad, facilidad de personalización (sistema de plantillas), capacidad de etiquetar recursos digitales, la organización de los elementos en colecciones y la importación y migración de datos mediante *CSV (Comma Separated Value)* y *OAI-PMH (Protocolo para la recolección de datos entre repositorios)* [22].



Logo 11.
Omeka

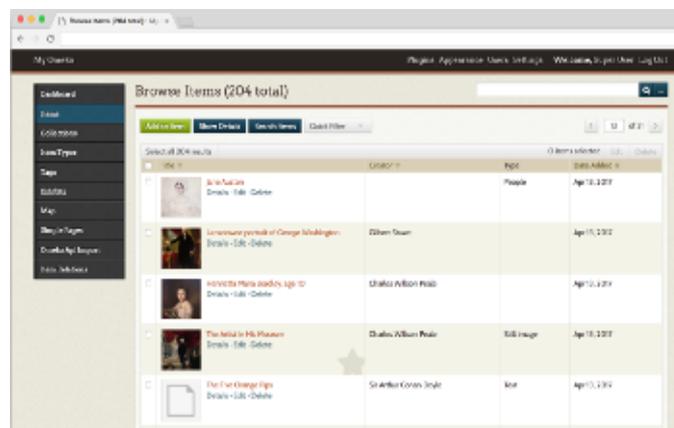


Figura 13. Repositorio con Omeka

Sin embargo, tras trabajar con él unos días y probar sus funciones, se decidió descartarlo por no ser lo suficientemente potente para organizar una gran cantidad de información de distinto tipo, ya que el solo hecho de disponer únicamente de colecciones haría que el repositorio estuviese mucho más desorganizado. Otra de las desventajas es la carencia de gestión de privilegios a usuarios, existiendo cuatro tipos de usuario predefinidos y sin la posibilidad de crear grupos.

Estos puntos débiles fueron las razones para no escogerlo como CMS del proyecto.

Joomla

Se decidió estudiar la posibilidad de utilizar *Joomla* como sistema de Gestión de Contenidos, debido a su popularidad (el segundo más utilizado del mundo) y la gran comunidad de desarrollo que lo respalda.

Joomla es un software gratuito de código abierto muy extendido, y utilizado en una amplia variedad de aplicaciones web; Blogs personales, aulas virtuales, foros venta de productos online etc.

Es un software de gran potencia y que no requiere de grandes destrezas a la hora de utilizarlo, y cuya administración tampoco es excesivamente complicada.

Cuenta con un gran número de extensiones y plantillas para poder personalizarlo, y en el caso de gestión de repositorios, se tuvo en cuenta Joomla ya que dispone de una extensión llamada *Remository*, que permite almacenar archivos remotos y permitir su acceso y descarga a los usuarios, así como personalizar la visualización de los archivos, entre otras características.

No obstante, no era lo suficientemente potente ni soportaba todas las funcionalidades descritas en los casos de uso, por lo que no se optó finalmente por *Joomla*.

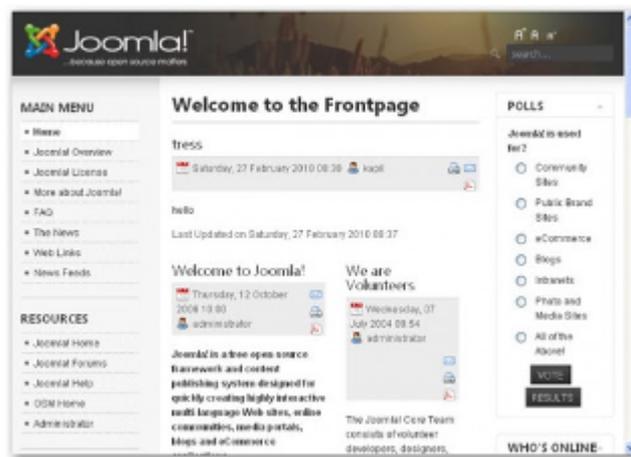


Figura 14. Ejemplo de página con Joomla

DSpace

Tras consultar en múltiples artículos, visitar varios repositorios digitales institucionales de entidades importantes (*Universidad de Oviedo, Universidad de Alicante, Universidad Católica Sedes Sapientiae* etc) y gracias al consejo de personal dedicado a la gestión del repositorio de la *Universidad de Extremadura*, se decidió por utilizar el CMS Dspace, el cual se detalla más en el apartado siguiente.

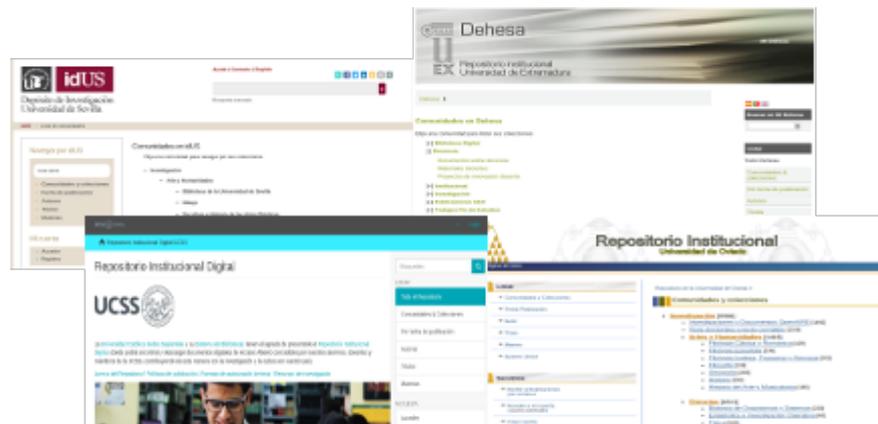


Figura 15. Repositorios Institucionales con DSpace

5.4.2 El CMS elegido para el proyecto: Dspace

Como se ha resaltado en el apartado anterior, se ha apostado para el desarrollo del proyecto de repositorio digital por el CMS *Dspace*.

Dspace es un software de código abierto desarrollado por el *Massachusetts Institute of Technology (MIT)* y la compañía *Hewlett Packard (HP)* enfocado a la gestión de repositorios digitales, tales como el que se desarrolla en este proyecto.

Es por esto que su uso está muy extendido entre instituciones académicas, como los repositorios insitucionales de la *Universidad de Extremadura (dehesa.unex.es)* , *Universidad de Oviedo (digibuo.uniovi.es/dspace/)* , *Universidad Católica Sede Sapientiae de Lima-Perú (repositorio.ucss.edu.pe)* , entre un amplio número de ejemplos de universidades, compañías etc.

Entre las principales señas de identidad de esta herramienta, cabe destacar: [23]

- Fácil acceso a diversos tipos de contenido digital (Texto, imágenes, audios..)
- Organiza los ficheros en Comunidades, divididas en Colecciones
 - Por ejemplo: Una Comunidad; *Facultad de Ciencias de la Salud*, dividida en Colecciones como *Enfermería, Farmacia, Medicina...*

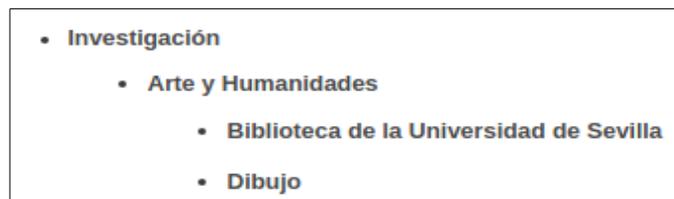


Figura 16. Comunidades y Colecciones (Univ. Sevilla)

- Cataloga los ficheros mediante metadatos (gracias al estándar *Dublin Core*)
 - Dublin Core fija un formato para la descripción semántica de los recursos de una web.
 - Los campos soportados para dicha tarea son 15, y vienen definidos de la forma *dc.[nombre del elemento]*.
 - Por ejemplo, para el título de una obra, se utiliza el campo *dc.title*

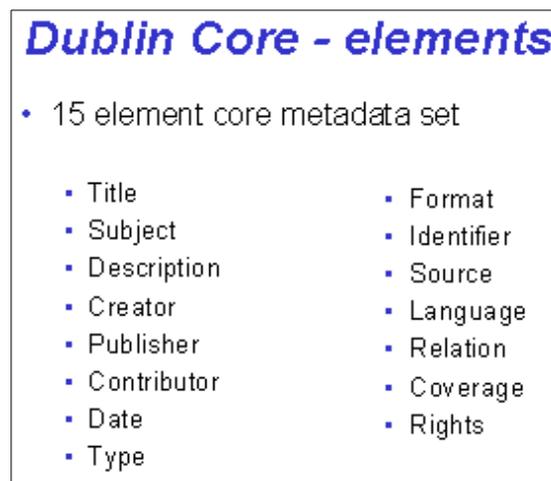


Figura 17. Campos definidos por Dublin Core

- Permite personalizar la apariencia y distintas funciones, como parámetros de búsqueda o de ordenación de los artículos.
- Permite soporte multilinguaje
- Permite la recolección de datos (o *Harvesting*) por parte de otras entidades por medio del protocolo *OAI-PMH*

DSpace se estructura en una arquitectura de 3 capas bien diferenciadas, cada una con una funcionalidad distinta:

- 1. Capa de Aplicación:** Es la capa superior, y en ella se encuentran la interfaz que interactúa con el usuario, así como otras herramientas auxiliares que añaden más funcionalidades (Estadísticas, exportación de datos, recolección de datos por parte de otras instituciones etc)
- 2. Capa de Negocio:** Se encarga del funcionamiento interno de *Dspace*, gestionando las búsquedas, los usuarios, la estructura del repositorio, los permisos etc.

- 3. Capa de Datos:** Como su nombre indica , se ocupa de gestionar los datos que se generan en el repositorio.
- Controla la Base de Datos, en la que se almacenan los metadatos de los artículos, los usuarios, la estructura del repositorio (comunidades y colecciones) etc.
 - Controla el almacenamiento de ficheros (o *Assetstore*), en el cual se almacenan los archivos subidos adjuntos a los artículos, el texto resultante de las indexaciones, las licencias etc.

El esquema que define esta arquitectura según la documentación oficial [24] es:

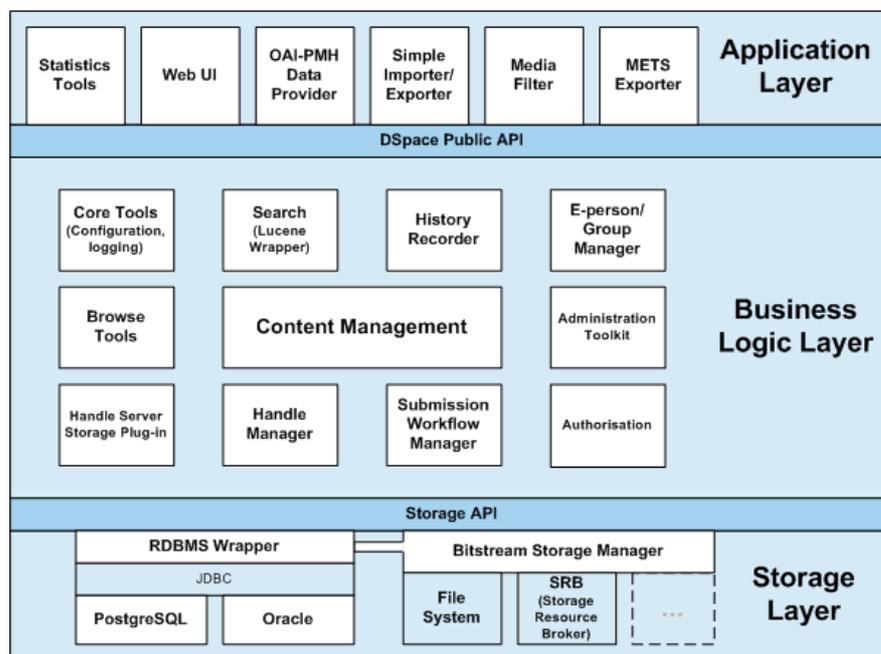


Figura 18.Arquitectura de DSpace

5.4.3 Servidor Web

Apache Tomcat

Apache Tomcat (anteriormente conocido como *Jakarta Tomcat*) es una aplicación de servidor web que soporta la implementación de *servlets* que son pequeños programas desarrolladas en Java instauradas en el servidor y que son llamadas desde el navegador web del cliente.



Logo 12. Apache Tomcat

Otra de las características de *Tomcat* es su capacidad de soportar las *Java Server Page (JSP)*, que son páginas dinámicas, es decir páginas HTML y XML con código Java incrustado

Utiliza para ello un compilador Java (*Jasper*) y un contenedor de *Servlets (Catalina)*.

Java Development Kit

Es un conjunto de herramientas de desarrollo para crear programas en lenguaje *Java*. Es una dependencia de *Tomcat*, por lo que debe haber instalado una versión de *JDK* en la máquina donde se aloje dicho servidor web.



Logo 13.
OpenJDK

5.4.4 Base de Datos

PostgreSQL

Es un Sistema Gestor de Base de Datos relacionales gratuito y de código abierto.

Una de las características a destacar es la escalabilidad, ya que no requiere usar bloques de lectura.

Está disponible para muchas plataformas y ofrece la herramienta *pgAdmin* para una gestión más simple.

Es el SGBD que utiliza *Dspace* por defecto.

PostgreSQL



Logo 14.
PostgreSQL

5.4.5 Sistema de Ficheros (Assetstore)

Se ha externalizado el almacenamiento de los datos del repositorio (ficheros, licencias etc) a un servicio de almacenamiento en la nube, por lo que se ha debido utilizar una herramienta y un protocolo determinado para este fin.

Nextcloud

Nextcloud es un servicio software dedicado a la sincronización y transferencia de ficheros de directorios locales a una cuenta de almacenamiento en la nube.

Proporciona programas tanto de la parte cliente como del servidor. En el servidor irá configurado un cliente para realizar la subida de ficheros.



Logo 15. Nextcloud

5.4.6 Herramientas del Sistema

Aplicaciones utilizadas para la administración del sistema.

Monit

Programa gratuito para la gestión y monitorización de distintos procesos, directorios, ficheros de sistemas UNIX. Permite controlar en tiempo real el estado de ciertos servicios, así como notificar al administrador en caso de fallos, o realizar ciertas acciones si sucede un determinado evento.



Logo 16. Monit

5.4.7 Lenguajes Utilizados

Si bien este proyecto no está enfocado al desarrollo, se han utilizado algunos lenguajes con otros fines.

XML

Lenguaje de marcado extensible, utilizado para representar de manera organizada la información de una web. Se ha utilizado a la hora de modificar algunas características de Dspace, por ejemplo para añadir un listado de vocabularios controlados.



Logo 17. XML

Java



Logo 18. Java

Como las funciones de Dspace vienen implementadas en este lenguaje, para modificarlas es preciso cambiar el código y recompilarlo.

Bash



Logo 19. Bash

Es un lenguaje de consolas de sistemas UNIX. Se ha utilizado en el servidor para realizar ciertos controles sobre la máquina.

6 DISEÑO

Tras la fase de análisis, en la que se han delimitado las funcionalidades que la herramienta va a ofrecer, se procede, pues, a realizar un diseño que defina la estructura del sistema, para su posterior implementación.

6.1 Arquitectura

En primer lugar, se ha definido la infraestructura que va a utilizarse para la implantación de la plataforma de repositorio web.

Para ello, se ha elaborado un esquema con los distintos componentes existentes, así como los actores que intervendrán en la utilización de la herramienta, y las interacciones entre todos ellos.

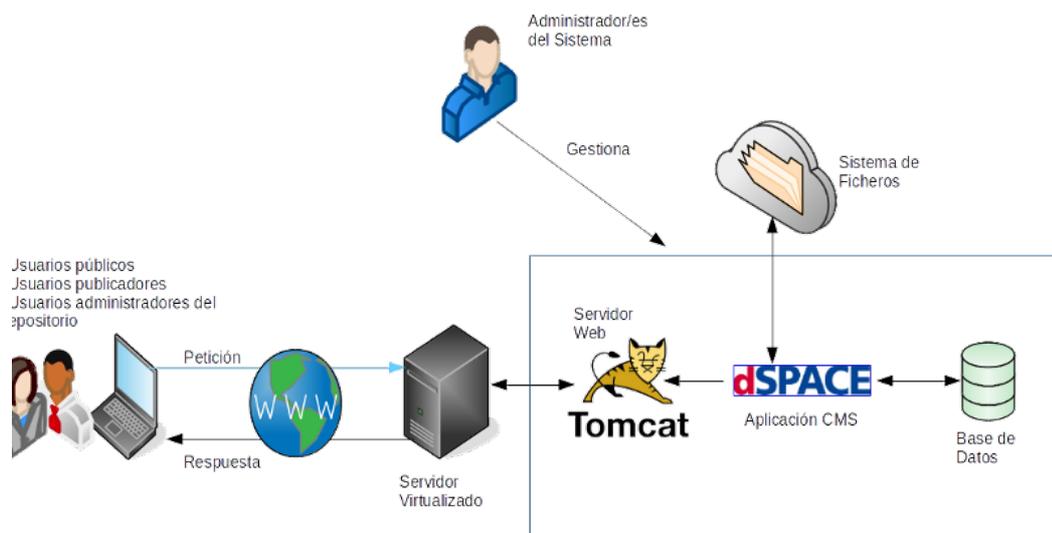


Figura 19. Arquitectura del Proyecto

Para comprender mejor el contenido del anterior esquema, se desarrolla la descripción de los componentes y del proceso que se sigue para la utilización de la web:

- Los **usuarios** (públicos, registrados y administradores) accederán al repositorio digital mediante su navegador web (en cualquier dispositivo: móvil, ordenador o tablet).

- El **servidor** que responde las peticiones de dichos usuarios es la máquina virtual antes referenciada.
 - En dicha máquina se encontrará el software de servidor web, que tendrá configurada como aplicación web el CMS usado como repositorio.
 - En la misma máquina se implanta una Base de Datos asociada al CMS, para almacenamiento de los usuarios, los esquemas de colecciones etc.
 - Se dispone de un sistema de ficheros externo, alojado en la nube, que será el encargado de almacenar todos los artículos publicados en el repositorio. De esta forma se asegura una mayor disponibilidad y una capacidad de almacenamiento mucho mayor.

6.2 Vista

Inicialmente se ha propuesto un modelo orientativo de la interfaz que tendrá la aplicación, en el que se refleja la disposición de los diferentes componentes.

Los componentes que se han incluido en este prototipo son a causa de considerarse de mayor relevancia para el usuario.

Se persigue no saturar en exceso de información, y de ofrecer una interfaz intuitiva

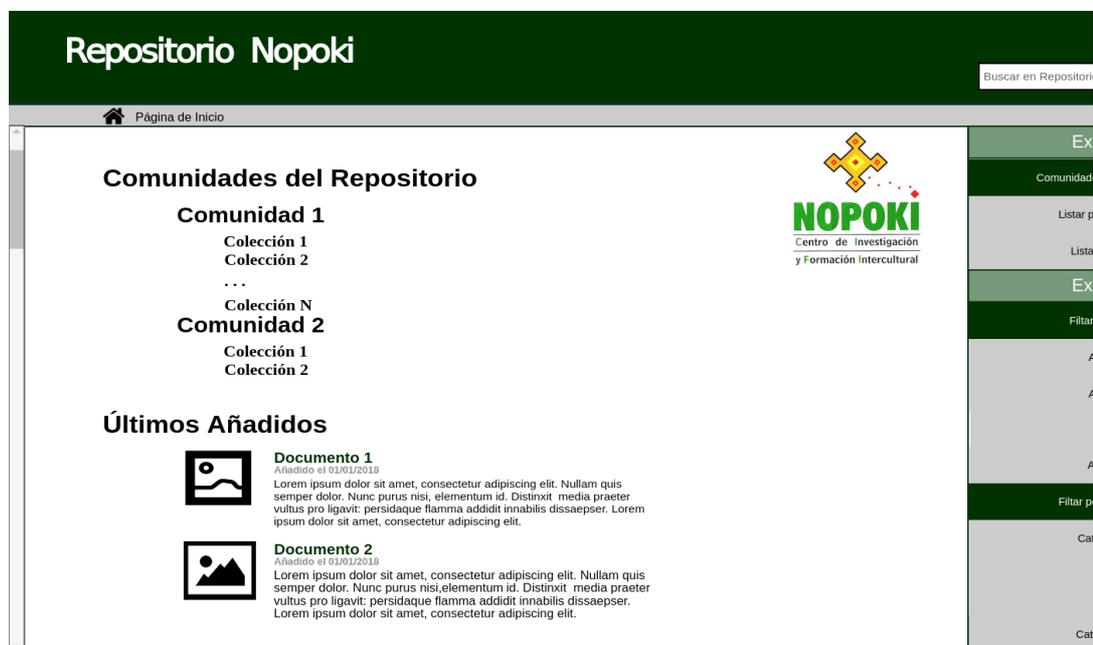


Figura 20. Prototipo de Interfaz

7 DESARROLLO E IMPLEMENTACIÓN

En este apartado séptimo se enumeran los pasos seguidos durante el despliegue de todo el sistema que da soporte al proyecto, sin entrar en detalles puramente técnicos, que irán descritos en posteriores anexos para el uso de futuros usuarios.

7.1 Configuración del Servidor

Herramientas instaladas y configuradas en el servidor virtualizado , así como sus dependencias, que sirvan como base del asentamiento de la plataforma web.

7.1.1 Apache Tomcat

El primer paso efectuado es la instalación de este servidor web en la máquina, que como se ha explicado es el necesario para que pueda funcionar *Dspace*, ya que sus funcionalidades están contenidas en *servlets* de *Java*.

Para que no haya problemas de compatibilidad, se ha debido de instalar una versión específica anterior a la última desarrollada, en concreto la versión 7. Como dicha versión se encuentra ya descatalogada en los repositorios oficiales de Ubuntu, se ha obtenido de un repositorio específico.

Tras su instalación, ha sido preciso configurar el servicio pertinente.

7.1.1.1 Java

Para que *Tomcat* pueda funcionar correctamente y compilar el código en *Java*, se debe tener instalado en el equipo servidor el conjunto de desarrollo de *Java* conocido como *JDK*, bien el ofrecido por la compañía *Oracles*, o bien la versión de código abierto *OpenJDK*, por la que se ha optado en este proyecto.

Al igual que ocurría con *Apache Tomcat*, la versión de *JDK* debe ser una versión específica (versión 1.7), por lo que también se ha obtenido por medio de repositorios externos.

7.1.2 PostgreSQL

Dspace utiliza como Sistema Gestor de Base de Datos *PostgreSQL* (también existe la posibilidad de utilizar *Oracle SQL*, pero tiene licencia propietaria).

Se ha instalado el servicio de *PostgreSQL*, para su posterior uso por el CMS.

7.1.3 Apache Maven

Maven es otra herramienta necesaria para poder instalar *Dspace*.



Logo 20. Apache Maven

Su función es la de construir proyectos *Java* (similar a *make* para proyectos en C) a partir de un fichero *XML* (*POM.xml*) sito en la carpeta raíz de dicho proyecto, el cual define la información sobre el proyecto (autor, identificador, dependencias etc).

Dichos ficheros *POM.xml* heredan información de otros, de manera que se puede centralizar información común a varios proyectos (dependencias, plugins habilitados, recursos etc).

Es el responsable de compilar y empaquetar todo el código *Java* , generando librerías (ficheros *.jar*) y aplicaciones web (ficheros *.war*).

La instalación de Apache Maven se hará mediante el repositorio de paquetes estándar de Ubuntu.

7.1.4 Apache Ant

Ant se encarga de realizar el despliegue de un proyecto *Java* previamente precompilado con *Maven*.



Logo 21. Apache Ant

De esta forma, su función es la de crear la estructura de directorios en la ruta donde se especifique, en base a un fichero *XML* (*build.xml*) que contendrá las rutas de origen del código a desplegar y las de destino de los directorios que se van a crear para albergar dicho código.

Se instala mediante el repositorio de paquetes de Ubuntu.

7.2 Instalación de DSpace

Una vez satisfechas todas las dependencias en la máquina para que pueda funcionar el CMS, se procede a realizar la fase de instalar el software de *Dspace* hasta conseguir tenerlo activo en el servidor y poder acceder a él mediante un navegador web.

7.2.1 Instalación en la máquina virtual

El proceso de instalación de *Dspace* conlleva una serie de pasos imprescindibles para su correcto funcionamiento.

1. **Creación de usuario en el servidor:** Se añade un nuevo usuario con el nombre *dspace* en la máquina de Ubuntu. Este usuario será el que utilice la aplicación para el acceso a determinados directorios y la ejecución de las herramientas *Java* sobre las que se apoya, por lo que tendrá unos determinados permisos sobre dichos recursos. El proceso de instalación deberá hacerse con ese usuario.
2. **Descarga del código fuente:** Se obtendrá el código fuente desde los servidores oficiales y se almacenará en el servidor. Es apropiado utilizar el código fuente si se requiere realizar una alteración de sus funcionalidades, si bien se podrían obtener los ficheros binarios, lo cual no es apropiado para este proyecto por sus limitaciones a la hora de personalizar la aplicación.
3. **Configurar la Base de Datos PostgreSQL**
 - 3.1 Creación de un usuario específico para *Dspace* para la Base de Datos con el nombre *dspace*.
 - 3.2 Creación de una nueva Base de Datos llamada *dspace*; cuyo propietario será el usuario creado previamente.
4. **Ajustar la configuración inicial:** Se establecen los parámetros básicos para instalar *Dspace*.
 - 4.1 Directorio de la instalación
 - 4.2 Nombre y URL en el cuál estará levantado el servicio (en nuestro caso *sight.unex.es/repositorio*)
 - 4.3 Nombre y usuario de la Base de Datos previamente creados para permitir su conexión
5. **Crear directorio de instalación:** Crear una carpeta donde se desplegará la aplicación, y darle la propiedad de dicha carpeta al usuario *dspace* de Ubuntu.
6. **Construir proyecto con Maven:** Desde el directorio donde se encuentran los ficheros fuente.
7. **Despliegue del proyecto con Ant:** En la primera instalación, *Ant* creará todo el árbol de directorios, desplegará el código, y creará el esquema de la Base de Datos *dspace*.

7.2.2 Configuración de Tomcat

Una vez completadas estas fases correctamente (cuando ha habido algún problema se ha vuelto a repetir el proceso completo), se tiene *Dspace* instalado en el servidor.

El siguiente paso será configurar el servidor *Tomcat*, que atiende las peticiones en Internet, para que sirva las distintas herramientas que componen *Dspace*.

7.2.2.1 Webapps de DSpace

Dspace se compone de distintas aplicaciones web o *webapps*, cada una con una funcionalidad específica. Existe por tanto la necesidad de configurar *Tomcat* para indicarle dónde se encuentran dichas *webapps*, por medio de los ficheros de configuración del servidor (no es una buena práctica copiar directamente los directorios de las *webapps* al directorio donde se encuentre instalado *Tomcat*).

A continuación se enumeran las *webapps* existentes para *Dspace* y su función, de una manera resumida.

JSPUI (Interfaz de Usuario JSP)

Es la interfaz basada en JSP , ya en desuso.

OAI (Iniciativa de Archivos Abiertos)

Interfaz encargada de la recolección (o *harvesting*) de metadatos por parte de autoridades específicas.

RDF

Extensión de *Dspace* capaz de exportar el contenido del repositorio a formato *RDF* (*Resource Description Framework*) para ser usado por lectores *RDF* como *SPARQL*.

REST

Aplicación basada en analizar peticiones mediante URLs.

SOLr

Apache SOLr es un potente y escalable motor de búsqueda que utiliza librerías del proyecto *Apache Lucene*, que proporcionan una amplia variedad de herramientas orientadas a las búsquedas.



Funciona como un servidor independiente implementado en Java. *SOLr*

Sus funciones en *Dspace* son las de buscar en textos completos, la indexación de documentos, la relevancia de términos o la elaboración de estadísticas.

Posee también una interfaz API Rest a través de la que se pueden hacer consultas.

SWORD y SWORDv2

Interfaces para recibir información mediante protocolo *SWORD (Simple Web-service Offering Repository Deposit)*, utilizado para transferir datos entre repositorios.

XMLUI (*Interfaz de Usuario XML*)

También llamada *Manakin*; es la actual interfaz web que utiliza *DSpace* en sus versiones más recientes. Permite mayor personalización que su antecesora (*JSPUI*), ya que se estructura en varias capas (Estilos, Temas y Aspectos). Se desarrolla su definición en próximos apartados.

Estas son, por tanto, las aplicaciones web que pueden ser utilizadas en un servicio *DSpace*, pero para el proyecto que nos ocupa no era necesario cargarlas todas en el servidor, por lo que se ha optado por solo contar con aquellas imprescindibles para soportar el funcionamiento del repositorio. Por este motivo, solo se han agregado tres aplicaciones: *OAI*, *SOLR* y *XMLUI*, aligerando así la carga de trabajo de *Tomcat* y posibilitando un acceso más rápido por parte del usuario.

7.2.3 Prueba de funcionamiento

Una vez todo instalado y configurado, se ha comprobado que la aplicación se comporta como debería y que se puede acceder a ella sin problemas.

Para probarlo, basta con acceder mediante un navegador web a la URL antes indicada que tendrá el repositorio (*sight.unex.es/repositorio*) a través del puerto por defecto de *Tomcat* (8080). Así, introduciendo en el navegador:

http://sight.unex.es:8080, aparece la siguiente pantalla de presentación:



Figura 21. Interfaz por defecto de DSpace

7.2.4 Configuración de DSpace

Tras haber instalado y comprobado que el CMS está correctamente corriendo sobre el servidor y se puede acceder desde cualquier dispositivo, se procede a la etapa más puramente técnica y que más tiempo ha requerido de todo el proyecto: Configurar la herramienta para adaptarlo a las necesidades del proyecto de *Repositorio Nopoki*.

7.2.4.1 Agrupamiento de documentos por categorías

La primera modificación consiste en ofrecer al usuario la capacidad de acceder a un determinado conjunto de artículos de un mismo tipo.

Para ello, se han definido una lista de los tipos más comunes de documentos que posiblemente se publiquen en el repositorio:

- **Animación**
- **Artículo**
- **Audio**
- **Documento**
- **Imagen**
- **Libro**
- **Literatura**
- **Presentación**
- **Tesis**
- **Vídeo**
- **Otro**

Tipo
Article [3]
Book [2]
Image [6]
Literature [1]
Other [1]
Thesis [4]

Figura 22. Agrupamiento por tipo

De esta forma, el usuario puede listar todos los documentos según una categoría. Por ejemplo, si un periodista busca imágenes de civilizaciones amazónicas, o si un doctorando quiere revisar tesis sobre una temática similar a la suya, pueden acceder a lo que están buscando de una manera más eficiente.

7.2.4.2 Sistema de etiquetado de documentos

Otro de los requisitos especificados para este repositorio es la capacidad de proporcionar un sistema de etiquetas o palabras clave que identifiquen a un determinado proyecto.

Para ello, se ha configurado un soporte que permita asignar una o varias etiquetas que el usuario elija y que estén relacionadas con el documento en cuestión.

Por ejemplo, si se va a publicar un documento que contiene una obra literaria escrita en el idioma peruano Quechúa, algunas etiquetas que podrían identificarlo podrían ser: *Literatura* , *quechúa* , *Perú*.

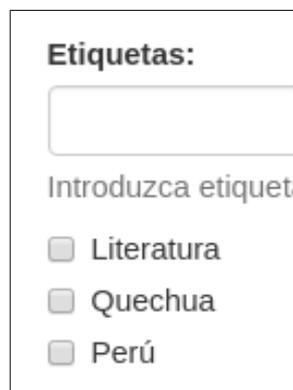
El formulario muestra el título "Etiquetas:" en negrita. Debajo hay un campo de entrada de texto vacío. Inmediatamente debajo del campo de texto, se lee "Introduzca etiquet". A continuación, hay tres opciones de selección con casillas de verificación: "Literatura", "Quechua" y "Perú".

Figura 23.Etiquetas

El usuario que publique dicha obra deberá asignarle esas etiquetas, facilitando su posterior acceso a dicho documento mediante alguna de esas etiquetas:

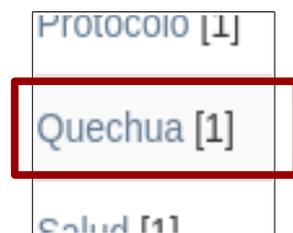
Una lista vertical de etiquetas. La etiqueta "Quechua [1]" está resaltada con un recuadro rojo. Las otras etiquetas visibles son "PROTOCOLO [1]" y "Salud [1]".

Figura 24.Etiquetas (2)

7.2.4.3 Indexación de documentos

Resulta de gran utilidad para poder acceder a una información específica el hecho de contar con la posibilidad de encontrar un documento que contenga texto con solamente introducir un fragmento de dicho texto.

Esta modificación permite extraer el texto plano de un documento digitalizado (en formato *.PDF*) así como de documentos de creados con aplicaciones de procesamiento de texto como *Microsoft Word* o *LibreOffice Writer*.

La función de indexado se hará mediante unos filtros específicos que *Dspace* proporciona, en este caso, los extractores de texto de PDF y Word.

7.2.4.3.1 Tarea de extracción de texto

Se ha implementado un pequeño *script* escrito en *Bash* en el servidor, que detecta la publicación de nuevos documentos en el repositorio, gracias a la herramienta *inotifywait* para Linux, que detecta cambios en un directorio, en este caso, en el directorio del *assetstore*.

Cuando un nuevo directorio se crea en el *assetstore* (almacenamiento de datos), aplica los filtros a los documentos del repositorio, saltándose aquellos en los que ya se hayan aplicado previamente. De esta manera siempre se indexarán los textos automáticamente al publicar un nuevo documento.

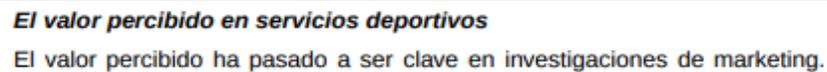
Desafortunadamente, la estructura de directorios del *assetstore* de *Dspace* no permite diferenciar la extensión del fichero subido, por lo que el filtro se aplica siempre, independientemente del tipo de archivo publicado, aunque el coste computacional no es elevado.

7.2.4.3.2 Ejemplo de búsqueda por texto contenido

En este pequeño ejemplo se observa la eficacia de este sistema de indexación.

Se procede a buscar el contenido de un documento *PDF* publicado en una de las colecciones del repositorio.

Este es un fragmento de texto contenido en un documento PDF correspondiente a un trabajo sobre deporte.



El valor percibido en servicios deportivos
El valor percibido ha pasado a ser clave en investigaciones de marketing.

Figura 25. Texto contenido en documento

Si introducimos este texto entrecomillado en el buscador general del repositorio:

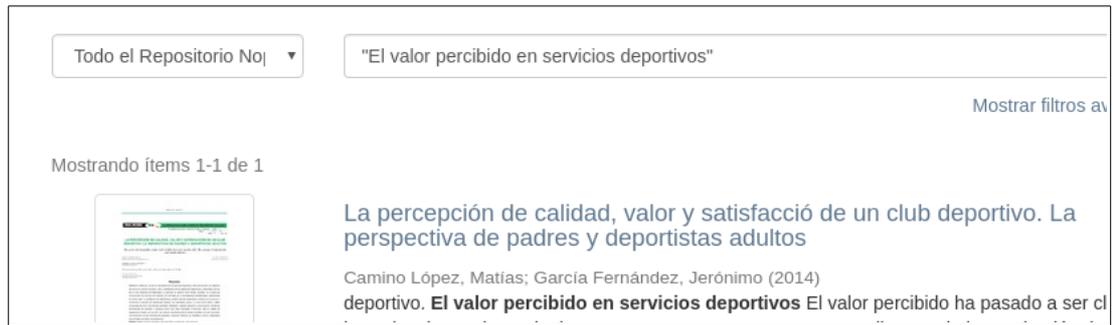


Figura 26. Resultado de búsqueda

Nos devolverá como resultado el artículo que contiene dicho texto en cuestión de milisegundos.

7.2.4.4 Personalización de los formularios de subida

Para la publicación de nuevos artículos, *Dspace* ofrece un formulario intuitivo y sencillo de usar en el que introducir los datos que caracterizarán dicho artículo (Autor/es, año de publicación, idioma, tipo, resúmenes sobre el contenido etc).

Sin embargo, el formulario por defecto cuenta quizá con demasiados elementos identificativos que no resultarán de relevancia al futuro usuario publicador del repositorio.

Es por ello que se han simplificado los formularios de subida y adaptado a las necesidades de él, y el formulario de envío de nuevos ítems se han ceñido a los siguientes campos, divididos en varias etapas (los campos con asterisco son obligatorios):

- **Etapas 1 – Describir ítem:**
 - Autor/es (pueden añadirse varios)
 - Título *
 - Títulos alternativos
 - Fecha de publicación * (al menos se debe introducir el año)
 - Editor
 - Identificadores: *ISSN, ISBN, ISMN, URI, Otro.*
 - Tipo: *Artículo, Audio, Documento etc* (Enumerados anteriormente)

- Idioma: *N/A (en caso de documentación no textual), Inglés, Inglés (USA), Español , Aimara, Quechúa, Otras lenguas del Perú, (Otras)*
- Etiquetas (Palabras clave para definir el documento)
- Resumen
- Descripción
- **Etapa 2 – Subir Fichero (Posibilidad de subir varios)**
 - Seleccionar archivo*
 - Descripción del Fichero
- **Etapa 3 – Revisión**
 - Resumen de los datos introducidos previamente y posibilidad de corregirlos
- **Etapa 4 – Licencia**
 - *DSpace* necesita del permiso del usuario publicador para aceptar la concesión de una Licencia *Creative-Commons* que permita la traducción, reproducción y envío por todo el mundo de la obra subida. Es necesario concederla si se quiere completar el envío.
- **Etapa 5 – Confirmar envío**

7.2.4.5 Configuración de los filtros de búsqueda

Otra modificación que se ha creído oportuna para adaptar el funcionamiento por defecto de *DSpace* al repositorio *Nopoki* es el de crear unos filtros de búsqueda avanzados según los parámetros que se prefieran.

Permiten acotar una búsqueda, por ejemplo, escribiendo uno de los apellidos del autor en el filtro de búsqueda por autor, el año de publicación en el filtro de fecha etc.

Se han realizado las configuraciones pertinentes para que el repositorio disponga de los siguientes filtros:

- **Título**
- **Autor**
- **Fecha**
- **Tipo**

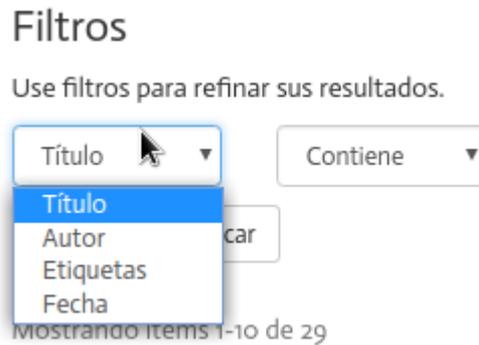


Figura 27. Filtros avanzados

7.2.4.6 Configuración de la apariencia

Se han llevado a cabo distintas acciones para ajustar el aspecto visual de la web del repositorio.

7.2.4.6.1 Interfaz de Usuario

De las dos interfaces que *DSpace* ofrece para su visualización, se ha optado por trabajar con la interfaz *XMLUI* (también recibe el nombre de *Manakin*).

Dicha interfaz ofrece una amplia variedad de posibilidades de personalización, ya que está construida en base a las herramientas del *framework Apache Cocoon*, empleadas en el desarrollo de sitios web basados en componentes, otorgándoles un carácter más modular y extensible.

Partiendo de esta base de separación de componentes, *XMLUI* se estructura en tres capas diferenciadas; cada una con un cometido específico:

- **Capa de Estilos (*Style Tier*):** En ella se define el aspecto puramente visual (hojas de estilo o *stylesheets* definidas en *CSS* e imágenes).
- **Capa de Temas (*Theme Tier*):** Para la definición de temas más complejos, añadiendo componentes o cambiando la estructura de la interfaz de usuario, por ejemplo. Se utilizan las librerías *XSLT (Extensible Stylesheet Language Transformations)* definiendo la estructura del tema, que servirán de base para generar los *XHTML* (archivos HTML escritos como XML) que se envían al cliente para que pueda visualizarlos. También en esta capa se incluyen archivos *CSS* y *JavaScript* para la apariencia final.

- **Capa de Aspecto (*Aspect Tier*):** Permite la creación de nuevas funciones a la interfaz mediante uso de clases Java.

Se han aprovechado algunas de las ventajas que reporta este modelo en capas para hacer algunas modificaciones respecto a la versión que viene por defecto, para adaptarlo a las necesidades del proyecto. Dichos cambios se detallan en los apartados siguientes.

7.2.4.6.2 Tema

La interfaz *XMLUI* ofrece varios temas predefinidos que se pueden aplicar al servicio web. Por defecto, con esta versión de *DSpace* se instala el tema llamado *Mirage*; sin embargo, se ha optado por la configuración de otro más actual y que ofrece mayor versatilidad.

Se ha instalado el tema *Mirage2*, de la compañía *Atmire*, que es a su vez proveedora de contenidos a la comunidad *DuraSpace*, encargada del desarrollo de *DSpace*.

Estas son las características que han determinado el uso final de dicho tema:

- **Adaptativo:** El diseño se adecua a distintas resoluciones, de tal manera que no varía cuando se redimensiona una ventana.
- **Fluido:** Escala el tamaño de las partes del diseño en relación al tamaño de la ventana.
- **Responsive:** El diseño se adapta a distintos dispositivos (móviles, tablets, ordenadores), optimizando el espacio disponible y mejorando la experiencia de usuario.

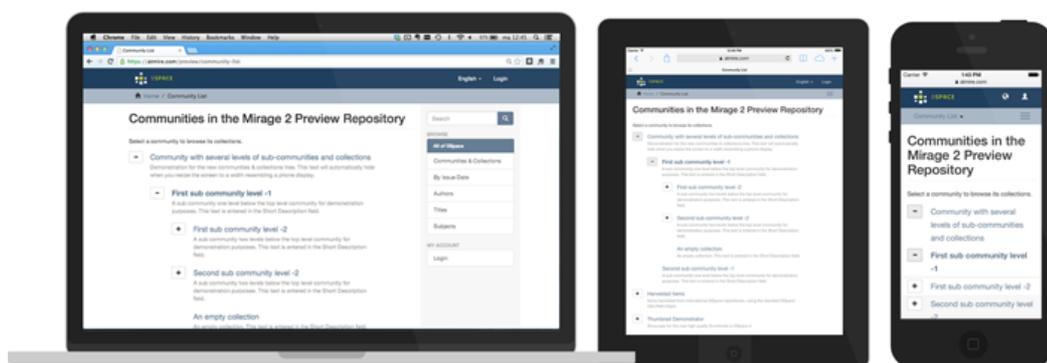


Figura 28. Tema *Mirage 2* en distintos dispositivos

Partiendo de la base de dicho tema, para la apariencia del repositorio *Nopoki* se han hecho cambios en la *Capa de Estilos* antes definida, para otorgarle un aspecto más personalizado y acorde con la organización a la que va destinada.

Estos cambios visuales han sido fundamentalmente, cambios en las hojas de estilos CSS para modificar la combinación de colores.

También se han editado los logotipos, añadido imágenes, una breve descripción en la cabecera, se han suprimido elementos que no se utilizan (p.ej: Sección de RSS), etc.

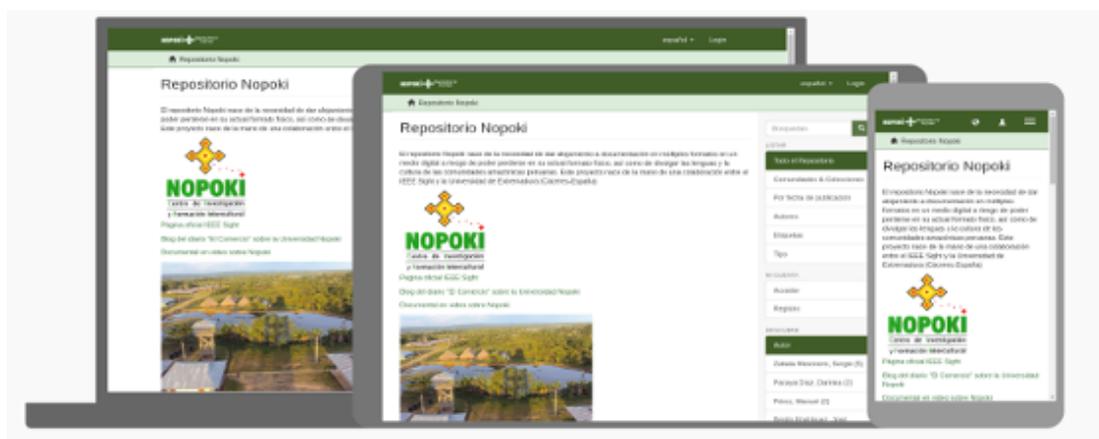


Figura 29. Repositorio *Nopoki* en distintos dispositivos

7.2.4.6.3 Internacionalización

Otro de los cambios desarrollados ha sido el de otorgar a la plataforma un soporte de múltiples idiomas.

Esto se ha conseguido gracias a la *internacionalización* o *i18n*, que consiste en:

- Asignar a cada componente de texto de la web un identificador único (p.ej: el campo *Comunidades & Colecciones* del panel lateral tendrá su propio identificador, que por poner un ejemplo será “*Navigation.communities_and_collections*”).
- Disponer de un fichero por cada idioma en el que se definan todos esos componentes de texto:
 - En el fichero que corresponde al idioma español, ese identificador “*Navigation.communities_and_collections*” tomará el valor “*Comunidades & Colecciones*”



Logo 23.
i18n

- En el fichero para el inglés; “*Navigation.communities_and_collections*” será “*Communities & Collections*”
- Y esto se repite para todos los idiomas.
- Definir un idioma o región local (en este caso el español).

Gracias a las facilidades que ofrece este sistema, se ha dado soporte para tres idiomas en el repositorio:

- **Español**
- **Inglés**
- **Portugués (de Brasil)**

7.2.4.6.4 Emails

Se han editado los mensajes de correo electrónico que la aplicación envía tanto a los usuarios que han realizado una acción (publicar un documento, cambiar su contraseña etc) como al administrador (nuevo usuario registrado, problemas internos del servidor etc), personalizándolos para el Repositorio Nopoki.

Este sería un ejemplo de esos correos, en este caso, para la verificación de que se ha publicado correctamente un documento:



Figura 30.Email de confirmación de publicación

7.2.4.7 Almacenamiento de Documentos

Terminada ya la etapa de configuración de la apariencia, se va a explicar en este apartado el sistema utilizado para el guardado de los ficheros con la documentación subida al repositorio.

7.2.4.7.1 Assetstore

Para el almacenamiento del contenidos de los ítems, *DSpace* utiliza un sistema de ficheros conocido como *Assetstore*.

En la arquitectura de *DSpace*, se encuentra en la capa inferior o *Capa de Almacenamiento*, en la que también se encuentra la Base de Datos.

Este *assetstore* es un directorio que contiene los ítems del repositorio identificados por un código numérico denominado *bitstream*, que indica en que subdirectorios se encuentran, por ejemplo: [25]

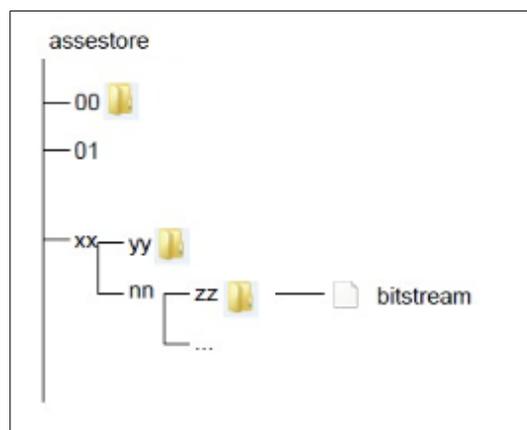


Figura 31. Estructura del assetstore

El patrón que seguirá el identificador de cada elemento siguiendo el esquema de arriba será:

xyyzznombre_del_fichero

Por tanto, un identificador *15022512345678910111213* nos indicará que ese ítem se encuentra en la ruta *assetstore/15/02/25/* y su nombre será *12345678910111213*.

Como podemos observar, en el identificador con que se almacena el documento, no se hace ninguna referencia a su extensión (*.pdf*, *.jpg*, *.doc* etc), ya que esa información se almacena en la base de datos.

7.2.4.7.2 Almacenamiento en la nube: Protocolo WebDAV

Si bien el antes mencionado *assetstore* (que no es más que un directorio) por defecto se crea como una carpeta en la propia máquina donde se haya instalado *DSpace*, se ha creído oportuno, con el fin de ganar en seguridad y en capacidad de almacenamiento, se ha optado por externalizar el *assetstore* a un entorno en la nube.

Se dispone de una cuenta de almacenamiento en la nube proporcionada por la *Escuela Politécnica* para el desarrollo de este trabajo:

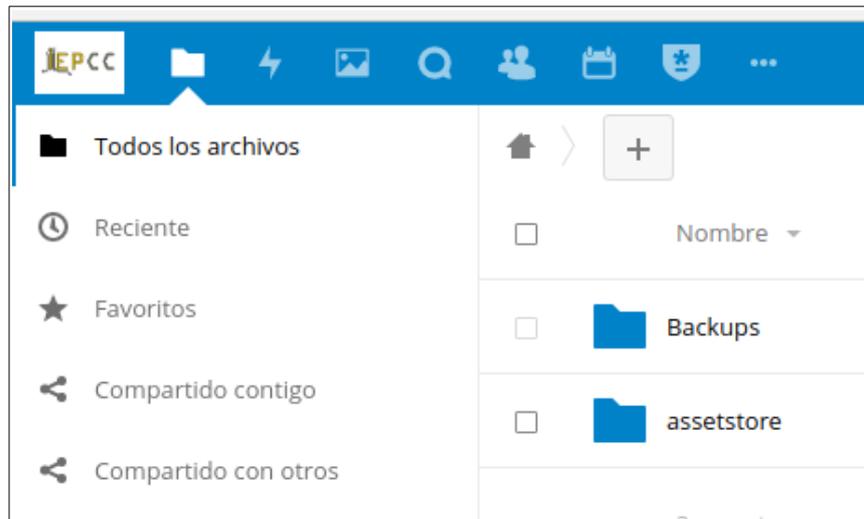


Figura 32. Servicio web para gestionar almacenamiento en la nube

Esta plataforma que nos ha sido proporcionada es un servidor *Nextcloud*, que es un conjunto de herramientas software *open source*, que permiten la instalación de un servicio de alojamiento en la nube en un servidor privado (similar a *Dropbox*).

Los pasos a seguir para el proceso de sincronizar los elementos subidos con dicha plataforma de almacenamiento en la nube han sido:

- 1. Configuración de un cliente Nextcloud:** La máquina virtual que aloja *DSpace* hará el rol de cliente, de manera que hay que configurar en ella el programa encargado de subir datos al servidor *Nextcloud* (la cuenta de almacenamiento proporcionada por la *EPCC*).
- 2. Montaje del sistema de ficheros remoto en un directorio local:** En la máquina virtual, se realiza un punto de montaje del sistema de ficheros remoto (la cuenta de almacenamiento de la nube) en un directorio local de esta (el directorio del *assetstore*), indicándole las credenciales del usuario que tiene acceso a dicha cuenta. El protocolo empleado para la transferencia de

archivos al servidor web es *WebDAV*, por lo que al realizar el montaje se debe indicar que el tipo de sistema de ficheros es *davfs*.

- 3. Comprobación del funcionamiento:** Se crea un archivo desde la máquina virtual en el directorio de *assetstore* y se comprueba desde el portal web que ofrece el servidor *Nextcloud* que aparece dicho archivo.

Como bien se ha apuntado en el paso 2, el protocolo utilizado para la transferencia al servidor Web es *WebDAV*, ó *Web-based Distributed Authoring and Versioning*.

Este protocolo se basa en la entrega de archivos que extiende las directivas HTTP existentes. La ventaja del protocolo es la seguridad, pues en este caso se usa HTTPS.

Por contra, es un sistema de transferencia de archivos lento e ineficiente, ya que en muchas ocasiones se pueden generar cuellos de botella en el servidor, debido al tráfico que genera *WebDAV*.

Existen otras alternativas de transferencia remota de archivos más eficaces como *NFS* o *SMB*, pero se ha adaptado el proyecto a los recursos disponibles.

7.2.5 Configuración del Servidor

En este nuevo apartado, se contemplan las actividades realizadas para la modificación e instalación de diferentes herramientas dentro de la máquina virtual para el correcto funcionamiento de todo el entorno del repositorio digital, así como de adición de algunas funcionalidades extra.

7.2.5.1 Procesos CRON

Dspace se vale de una serie de trabajos o *jobs* que se ejecutan de una forma periódica y programada para lograr un correcto funcionamiento y estar actualizado. Dichos trabajos se planifican en los sistemas tipo *UNIX*, como en el caso de la máquina *Ubuntu* mediante procesos *cron*, que son procesos que corren en segundo plano con la periodicidad que se les indique. Estos procesos se listan en un fichero específico de *UNIX*, llamado *crontab*.

```
0 6 * * 0 $DSPACE/bin/dspace checker-emailer
```

Figura 33. Ejemplo de tarea CRON

Esta es la lista de las tareas de *Dspace* realizadas en segundo plano:

- **Tareas diarias**
 - Regenerar los índices de SOLR, limpiando aquellos que se hayan borrado, y creando los nuevos

- Mantenimiento de las estadísticas y generación de las estadísticas diarias
- Notificar de la creación de nuevos contenidos en determinada colección a los usuarios que estén suscritos a esta.
- Tareas de curación (antivirus, conversión de formatos...)
- **Tareas semanales**
 - Verificar los *checksums* de los ficheros almacenados en el repositorio, para detectar cambios o estados erróneos y notificación al administración de dichas comprobaciones.
- **Tareas mensuales**
 - Borrado permanente de ficheros marcados como borrados en el repositorio, pero que siguen existiendo en el directorio de ficheros.
 - Actualización del fichero de *geolite*, que identifica la localización geográfica de los accesos a los elementos del repositorio (Por ejemplo se emplea para mostrar las estadísticas de visitas por país o ciudad).
- **Tareas anuales**
 - Reiniciar estadísticas anuales
- También se indican algunas variables globales como el directorio de *DSpace* o la memoria asignada a Java, por ejemplo.

7.2.5.2 Configuración de HTTPs

Por defecto, el servicio web que ofrece *Apache Tomcat* se sirve mediante el protocolo *HTTP*, con el inconveniente de que el contenido se transfiere por la red sin ningún tipo de encriptación, y en él pueden viajar datos relevantes como credenciales.

Es por ello que se ha optado por habilitar el uso de certificados *SSL (Secure Sockkets Layer)*, para permitir el uso de *HTTPS (HTTP Secure)*, y que exista una encriptación de los datos transferidos.

Para ello, se ha obtenido un certificado *SSL* de manera gratuita desde la plataforma [Let's Encrypt](#), que ofrece certificados de manera gratuita y verificados por Autoridades de Certificación (CA), válido para un dominio específico (en este caso; *sight.unex.es*). La duración de este certificado es de 3 meses, no obstante se puede renovar sin coste adicional.

Una vez obtenido el certificado, se configurará en el servidor *Tomcat* para que escuche por el puerto asignado a *HTTPS* (443) y redireccione las peticiones que vengan del puerto *HTTP* a dicho puerto 443.

Una vez configurado y reiniciado el servicio, ya estará habilitado el acceso seguro a la página, mediante la URL <https://sight.unex> (si se introduce la URL <http://sight.unex.es>, se redireccionará al puerto *HTTPS*).



Figura 34. Certificado SSL habilitado

7.2.5.3 Copias de seguridad

Se han implementado políticas de realización de copias de seguridad de los componentes fundamentales del proyecto para su recuperación en caso de posibles fallos.

7.2.5.3.1 Base de Datos

Se ha diseñado un pequeño programa que gestiona las *backups* de la base de datos de *DSpace*, almacenándolas en un medio externo (cuenta en la nube).

La política que se ha definido es la de realizar un volcado del contenido de la base de datos diario y almacenar las copias de la última semana, borrando la más antigua cuando se inserta una nueva.

Esto se ha realizado mediante un *script* desarrollado en *Bash*, incluido en el *crontab* y que se ejecuta con periodicidad diaria.

7.2.5.3.2 Assetstore

Para disponer de un respaldo en los ficheros almacenados en el repositorio, diariamente se realiza una copia de seguridad del directorio de ficheros.

Inicialmente se comprime dicho directorio y se guarda en un medio externo (para el prototipo del proyecto, se ha empleado el propio equipo de trabajo).

Como en la actualidad no se dispone de demasiado almacenamiento, provisionalmente se almacenarán sólo las copias de los dos últimos días.

7.2.5.3.3 Ficheros de configuración

Se ha creído conveniente salvar las configuraciones más relevantes del servidor, en caso de necesidad de una reinstalación de todo el sistema o para futuras migraciones.

Los ficheros que se han almacenado son:

- **Configuraciones de Tomcat**
- **Configuraciones de DSpace**
- **Ficheros de UNIX (Configuración de los sistemas de ficheros *fstab*)**
- **Scripts**

Estos ficheros se almacenan manualmente cuando se hayan realizado cambios relevantes y comprobado que funcionan correctamente, por lo que los ficheros que se encuentran en el medio de almacenamiento externo serán las últimas versiones correctas.

El medio externo para almacenar estas configuraciones ha sido la cuenta en la nube.

7.2.5.3.4 Logs

Los *logs* generados por la actividad de *DSpace* así como los generados por los *scripts* creados para el proyecto, se almacenarán en la nube.

7.2.5.4 Monitorización

Para una administración más eficaz de todo el sistema, y tener un control en tiempo real de todo el servidor, se ha implantado un entorno de monitorización.

Esto se ha conseguido mediante la instalación y configuración de una herramienta software para *UNIX* llamada *Monit*.

Esta herramienta permite controlar el estado de distintos tipos de recursos (procesos, sistemas de ficheros, parámetros del Equipo como el % de memoria usado etc), y designar tareas asociadas a que ocurra un determinado evento (p.ej: Enviar un email en caso de que un proceso falle, reiniciar el equipo si se excede una cantidad de memoria etc).

En el servidor se han configurado estos recursos y eventos:

- **Recursos monitorizados:**
 - Máquina virtual
 - Proceso de servidor *tomcat*
 - Proceso de servidor *openssh*
 - Proceso de servicio *postgresql*
 - Sistema de Ficheros del *assetstore*

```
## Check Apache Tomcat Status
check process tomcat7 with pidfile /opt/tomcat/temp/tomcat.pid
  start program = "/usr/sbin/service tomcat start" with timeout 50 seconds
  stop program = "/usr/sbin/service tomcat stop"
  if cpu > 60% for 2 cycles then alert
  if cpu > 90% for 5 cycles then restart
  if failed port 80 then alert
```

Figura 35. Configuración de monitorización de Tomcat

- **Eventos definidos**
 - Si uso del CPU de la máquina o la memoria usada superan el 80 % alertar por email
 - Si alguno de los procesos (*postgresql*, *tomcat*, *openssh*) falla, se alerta por email y se reinicia dicho servicio
 - Si se sobrepasa el 80% del almacenamiento disponible del *assetstore* alertar por email.

Esto será de ayuda para garantizar una mayor disponibilidad del servicio.

Monit Service Manager						
Monit is running on sight.unex.es with uptime, 23h 28m and monitoring:						
System	Status	Load	CPU	Memory		
sight.unex.es	Running	[0.05] [0.29] [0.30]	0.3%us, 0.3%sy, 0.0%wa	9.3% [760.1 MB]	0.0%	
Process	Status	Uptime	CPU Total	Memc		
tomcat7	Running	23h 16m	0.0%	6.3% [
ssh	Running	10d 1h 9m	0.0%	0.4%		
postgresql	Running	10d 1h 9m	0.0%	2.3% [
Filesystem	Status	Space usage	Inode			
Assetstore	Accessible	0.4% [75.2 MB]	0.0%	4%		

Figura 36. Servicio web de Monit

7.2.5.5 Optimización del rendimiento del servidor

Apache Tomcat necesita muchas dependencias para ofrecer el servicio, por lo que se han llevado a cabo algunas medidas para reducir su carga de trabajo y aumentar la velocidad de acceso al repositorio, a fin de ofrecer una mejor experiencia de usuario.

Este es el listado de las tareas llevadas a cabo para aumentar el rendimiento del servidor:

- Eliminar las librerías de *websockets*
- Eliminar el escaneo de ciertos *servlets*
- Incluir en *Tomcat* sólo aquellas aplicaciones web de *DSpace* que vayan a ser utilizadas (*XmlUI*, *SOLR* y *OAI*).

Este proceso reduce el tiempo de acceso a la plataforma web de más de 40 segundos a menos de la mitad.

7.2.6 Gestión de Usuarios

En esta sección se explica el proceso de creación de usuarios, así como los tipos de usuarios que existen y las acciones y privilegios de que dispone cada uno.

7.2.6.1 Usuario Administrador

El usuario administrador será el que tenga el control sobre todos los ámbitos del repositorio. Así pues, un administrador tendrá estas capacidades:

- Gestión de Comunidades y Colecciones
 - Creación, edición y eliminación
- Gestión de los Usuarios
 - Listar, añadir y eliminar usuarios.
- Gestión de los Grupos
 - Creación, edición y eliminación de Grupos
 - Agregación y supresión de Usuarios y/o Grupos a un Grupo
- Gestión de los Privilegios
 - Crear privilegios (Lectura, Creación, Eliminación, Edición y Administración) sobre cada Comunidad y/o Colección.
 - Asignación de dichos privilegios a uno o varios Grupos de usuarios previamente creados.
- Importación de Ítems mediante ficheros externos
 - Importar metadatos de un ítem

- Importar fichero *.zip* que contenga un ítem
- Visualización de Estadísticas sobre el repositorio
 - N.º de elementos guardados
 - Visitas a colecciones, comunidades, ítems
 - N.º de accesos de usuarios
 - N.º de búsquedas totales
 - *Etcétera*
- Control de los Envíos
 - Envíos no terminados (que puede completar)
 - Envíos completados
- Otras acciones
 - Gestión de los formatos de fichero soportados (*.doc, .jpg, .pdf etc*)
 - Gestión de los esquemas de metadatos soportados (*Dublin Core, EPerson etc*)

Este tipo de usuario será creado desde el propio servidor por el administrador del sistema.

7.2.6.2 Usuario Publicador

Es el otro tipo de usuario, y los permisos que tenga vendrán definidos por el Usuario Administrador.

Un visitante podrá registrarse desde el sitio web mediante sus datos personales (*correo electrónico, Nombre y Apellidos etc*) y por defecto no tendrá más permisos que la visualización de los contenidos y de las estadísticas básicas.

El usuario deberá solicitarle al administrador (mediante el formulario de contacto) la capacidad de publicar documentación en una o varias comunidades o colecciones según la naturaleza de esta, y este podrá incluirlo o no en un grupo de publicadores que tenga permisos sobre una o varias determinadas comunidades y/o colecciones.

Una vez que se le incluya en un grupo, el administrador deberá notificárselo por email para que conozca la nueva situación y pueda empezar a publicar la documentación.

Si el usuario publicador infringe las normas publicando artículos que no cumplan las normas podrá ser eliminado por el administrador.

8 RESULTADOS Y PRUEBAS

Tras finalizar toda la fase del desarrollo de la plataforma, se ha contrastado el funcionamiento de esta evaluando todas y cada una de las funcionalidades que ofrece.

El resultado ha sido satisfactorio, obteniendo un repositorio digital muy práctico que cumple con los requisitos funcionales previstos con un buen rendimiento, y además ofreciendo una sencillez de utilización, debido a que está enfocado fundamentalmente a personal que no necesariamente dispone de conocimientos técnicos.

En definitiva, se estima que la solución desarrollada entra dentro de la idea inicial que solicitaba la comunidad *Nopoki*, ya que se ha procurado adaptarlo a la filosofía y necesidades de esta, y está listo para empezar a ser utilizado.

Cabe destacar también que el repositorio podrá, en un futuro, ser trasladado a otro entorno y ser administrado por otra persona o grupo, por lo cual se adjunta una guía con los pasos para tal fin.

8.1 Pruebas realizadas

En cuanto a las comprobaciones que se han ido efectuando durante todo el desarrollo del proyecto, no se ha seguido ninguna metodología específica, sin embargo, se ha fijado la norma de probar cada modificación o funcionalidad realizada, y no continuar con el progreso de la aplicación hasta que se haya verificado su correcto funcionamiento.

Para las pruebas del repositorio, se han ido creando Comunidades y Colecciones de ejemplo, dentro de las cuales se han ido introduciendo distintos documentos de múltiples formatos, para realizar sobre ellas pruebas de búsqueda, ordenación, filtrados, indexación etc.

8.2 Problemas surgidos

Han sido varios los conflictos que se han generado durante toda la implementación debido a que se trabaja con múltiples tecnologías ensambladas entre sí.

Estos problemas suponen un recargo importante del tiempo dedicado al proyecto, ya que suponen un exhaustivo análisis de las posibles causas, así como la búsqueda de información en distintas fuentes hasta dar con una solución.

Se realiza una enumeración de los principales problemas que han acaecido:

- **Dificultad de implantación de HTTPS:** Se han dado múltiples errores a la hora de configurar el protocolo seguro *HTTPS* en el servidor, puesto que esto generaba un conflicto para el motor de búsqueda *Apache SOLR*, el cuál solo trabaja con directivas *HTTP*. La solución ha sido de dar soporte *HTTPS* a todo el servidor y añadir la excepción de la aplicación web de *SOLR*, lo cuál no supone un riesgo ya que su acceso está limitado solo a la máquina virtual.
- **Falta de memoria:** La compilación e instalación de *DSpace* en la máquina supone un alto consumo de la memoria RAM de ésta, por lo que fue necesario solicitar un aumento de ella al Servicio Técnico de la Escuela Politécnica, de 2 a 8 GB.
- **Problemas con caracteres especiales:** No se permitía la búsqueda de campos que contuviesen acentos o caracteres especiales (Apellidos, Etiquetas etc), por lo que fue necesario habilitar la codificación *UTF-8* en el servidor *Tomcat*.
- **Otros problemas:** Problemas con los *scripts* creados, problemas de acceso a la máquina, errores al añadir nuevos elementos como filtros de búsqueda etc.

9 CONCLUSIONES

A continuación se comentan las principales conclusiones derivadas de la realización del presente proyecto.

En lo que concierne al ámbito cooperativo y humanitario, la principal conclusión obtenida es la gran utilidad que este repositorio puede suponer para facilitar la divulgación del patrimonio cultural de las comunidades de la Amazonía peruana, preservando este legado y ayudando a su estudio por expertos de todo el mundo.

Si bien este proyecto no realiza una intervención directa sobre una población local, como puede ser la instalación de una red eléctrica o de saneamiento en un área rural de un país en vías de desarrollo, este proyecto repercute de manera indirecta sobre estas poblaciones, ya que ayuda a conocer su situación y a que más personas interesadas puedan involucrarse en acciones de este tipo. De esta forma, se propicia seguir avanzando en la dirección de mejorar la calidad de vida de regiones que actualmente no cuentan con los suficientes recursos técnicos y humanos cualificados de los que disponemos en nuestro país.

En el ámbito personal, el proyecto me ha reportado una gran satisfacción por el hecho de haber colaborado con una asociación de tanta relevancia como *IEEE*, formando parte de una red global de técnicos que aportan su conocimiento de forma voluntaria y desinteresada para el desarrollo de la sociedad.

Tras finalizar este proyecto, se ha generado un firme interés por continuar realizando proyectos de este tipo, así como de ingresar como miembro del *IEEE Sight (Grupos de Especial Interés en Tecnología Humanitaria)*.

En el plano puramente técnico, la principal conclusión radica en la actualización de muchos conocimientos útiles para la capacitación profesional en el ámbito informático, pues ha sido un proyecto transversal que abarca un amplio conjunto de tecnologías modernas en los campos de *servidores web, bases de datos, motores de búsqueda, sistemas de ficheros, etc.*

En este sentido, ha sido preciso realizar una ardua labor de documentación para el manejo y configuración de la herramienta *DSpace* que, si bien es muy potente, requiere de altos conocimientos técnicos sobre ella. No en vano, existen compañías consultoras dedicadas específicamente para su administración.

Es importante también destacar la de adquisición y experiencia en la capacidad de planificar tareas en un proyecto de desarrollo amplio que involucra distintas tecnologías, fijando los distintos objetivos que deben ser desarrollados a corto plazo.

Finalmente, en cuanto al análisis de las posibilidades de desarrollo futuro del Repositorio Digital *Nopoki*, mi estimación es que este repositorio podrá ser aprovechado tal como está su estado actual para comenzar a trabajar en el almacenamiento de grandes cantidades de documentación. No obstante, la posibilidad de añadir nuevas funcionalidades y mejorar las ya existentes son muy amplias y permiten enfocar futuras propuestas de desarrollo.

10 TRABAJOS FUTUROS

Se han planteado algunas posibles extensiones o mejoras que pueden ser útiles para añadir a la plataforma.

10.1 Servicio OAI

Una utilidad sumamente interesante sería la de poder activar el servicio OAI para ofrecer la posibilidad de que los datos albergados en el repositorio puedan formar parte de un directorio común de objetos digitales a nivel de Perú o de Latinoamérica.

Para ello, se debería utilizar el protocolo *OAI-PMH (Open Archive Initiative-Protocol for Metadata Harvesting)*, dedicado a la transferencia de metadatos con un formato estándar.

El proceso que se sigue en esta funcionalidad de *Harvesting* o *Recolección de datos* es que un cliente Recolector *OAI-PMH* (p.ej: *El Ministerio de Cultura*) realiza peticiones de metadatos a nuestro repositorio, que actúa como *Proveedor de Datos*, y que deberá tener el servicio *OAI* habilitado. De esta forma, el *Recolector* podrá obtener nuestros objetos digitales y que sean accesibles desde una colección común.

En España y Europa existen algunos ejemplos de estas colecciones, como por ejemplo la colección *Hispana* o *Europeana*.

- *Más información sobre Hispana y Europeana:*
 - <http://hispana.mcu.es/>
 - <https://pro.europeana.eu/>

10.2 Migración de servicios a la nube

Sería altamente recomendable el traslado de todo el repositorio digital, o al menos algunos de sus componentes más importantes (como la Base de Datos o el directorio de almacenamiento) a un medio externo que ofrezca mayor disponibilidad y capacidad de cómputo y/o almacenamiento. Algunos de estos medios externos son *Amazon Web Services* o *Microsoft Azure*.

10.3 Funcionalidad OCR

La integración de un módulo de conversión digital mediante *OCR (Reconocimiento óptico de Caracteres)* podría añadir funcionalidades muy útiles al repositorio.

La funcionalidad OCR se planteó inicialmente como posible requisito del proyecto. No obstante, debido al alto coste económico del software dedicado a este fin, y a la complejidad que supondría para documentos en ciertas lenguas peruanas, se descartó para esta etapa inicial, dejándola como posibilidad futura.

Existen soluciones para acoplarla la funcionalidad OCR a *DSpace*, como la desarrollada por la compañía *4Science*. Su alto precio (3.000 €) fue, como hemos dicho, la principal razón para desestimar esta funcionalidad en el presente trabajo.

11 BIBLIOGRAFÍA

- [1] UNIVERSIDAD CATÓLICA SEDES SAPIENTAE. “NOPOKI, educación de los pueblos indígenas” <http://camp.ucss.edu.pe/blog/exposicion-nopoki-educacion-de-los-pueblos-indigenas>
- [2] EL COMERCIO. “NOPOKI:La universidad de los indígenas” <https://elcomercio.pe/blog/heroesantesde30/2014/03/nopoki-la-universidad-de-los-indigenas>
- [3] IEEE SIGHT. *IEEE SIGHT Projects* <http://sight.ieee.org/about-ieee-sight/ieee-sight-projects/>
- [4] 1&1 DIGITAL GUIDE. *Servidor web: definición, historia y programas* <https://www.1and1.es/digitalguide/servidores/know-how/servidor-web-definicion-historia-y-programas/>
- [5] ECURED. *Servidor Web* https://www.ecured.cu/Servidor_Web
- [6] REPRESAS,Carlos. *Funcionamiento de un Servidor Web* <https://codigointegral.wordpress.com/2015/09/25/funcionamiento-de-un-servidor-web/>
- [7] SUÁREZ FALCÓN, YANIEL. *Funcionamiento de un Servidor Web* <https://deprogramacion.cubava.cu/2016/01/14/funcionamiento-de-un-servidor-web/>
- [8] UBUNTU DOCUMENTATION. *HTTPD* <https://help.ubuntu.com/lts/serverguide/httpd.html>
- [9] TECHOPEDIA. *Apache Tomcat* <https://www.techopedia.com/definition/15735/apache-tomcat>
- [10] ORACLE. *JavaServer Pages Technology* <https://www.oracle.com/technetwork/java/index-jsp-138231.html>
- [11] KINSTA. *¿Qué es NGINX y cómo funciona?* <https://kinsta.com/es/base-de-conocimiento/que-es-nginx/>
- [12] ROUSE, Margaret. *Internet Information Services (IIS)* <https://searchwindowsserver.techtarget.com/definition/IIS>
- [13] WEBTEMATICA. *Qué es la web semántica* <https://webtematica.com/que-es-la-web-semantica>

- [14] DELGADO, Hugo. *Web Semántica, definición, historia y características*
<https://disenowebakus.net/semantica-web.php>
- [15] BENGOCHEA RUÍZ, María Jesús. *La Web Semántica o Web 3.0*
<https://www.orientadorweb.com/web-semantica/>
- [16] LAMARCA LAPUENTE, María Jesús. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen.*
<http://www.hipertexto.info/documentos/rdf.htm>
- [17] UNAM. *RDF y RDF schema*
http://www.matem.unam.mx/~grecia/semantic_web/rdf.html
- [18] DIARIO ABC. *Las diez web semánticas que han sorprendido en 2010*
<https://www.abc.es/20101213/medios-redes/webs-semanticas-2010-201012131134.html>
- [19] KINSTA. *Content Management System*
<https://kinsta.com/knowledgebase/content-management-system>
- [20] FERNÁNDEZ ALONSO, Antonio. *¿Qué es un CMS? Conoce los mejores gestores de contenido* <https://www.webempresa.com/blog/que-es-cms-los-mejores-gestores-de-contenido.html>
- [21] INTERNETYA. *¿Qué es un CMS y qué ventajas tiene?*
<https://www.internetya.co/que-es-un-cms-y-que-ventajas-tiene/>
- [22] ALCARAZ, Rubén. *Omeka 2* <http://www.rubenalcara.es/manual-omeka/introduccion-omeka.html>
- [23] SANZ, Emilio. *¿Por qué implantar DSpace para la gestión de repositorios digitales en instituciones, organizaciones y empresas?*
<http://sorprendemos.com/consultoresdocumentales/?p=6395>
- [24] DURASPACE. *DSpace 6.X Documentation*
<https://wiki.duraspace.org/display/DSDOC6x/Architecture>
- [25] LORENZO, Emilio. *Hablando de DSpace* <https://www.arvo.es/dspace/la-estructura-del-assetstore/>

ANEXOS

ANEXO I - Guía de Usuario

AI.1 Descripción de la Aplicación

El *Repositorio Nopoki* es una plataforma web ideada para el almacenamiento masivo de documentación digital relacionada con las comunidades indígenas que conforman la *Universidad Bilingüe Nopoki*, con sede en *Ucayali (Perú)*.

Su objetivo es el de la preservación y divulgación de aquellas culturas y lenguas, como el *asheninka* o el *matchiguenga*, en riesgo de desaparición.

La idea inicial nace de una cooperación internacional entre el *IEEE Sight* y la *Universidad de Extremadura* (España).

AI.2 Requisitos

Para utilizar el repositorio solo se necesita un equipo (ordenador, móvil o tablet), acceso a Internet y un navegador web que soporte *HTML5* y *JavaScript*.

AI.3 Manual de Usuario

AI.3.1 Acceso al Repositorio Digital

Acceder mediante el navegador web a la URL <https://sight.unex.es/repositorio>

AI.3.2 Visualización de Contenido

Se puede explorar el contenido del repositorio de diversas formas según la naturaleza o procedencia de la documentación:



Todo el Repositorio
Comunidades & Colecciones
Por fecha de publicación
Autores
Títulos
Etiquetas
Tipo

Figura 37. Panel "Listar"

- **Por Comunidades y Colecciones**
 - **Comunidades:** Son la categoría mayor en que se estructura el repositorio, y hacen alusión a ámbitos generales (p.ej: *Documentación académica, Investigación, Literatura etc*). Las Comunidades se estructuran en Colecciones, e incluso pueden incluir Comunidades que se denominarán *Subcomunidades*.
 - **Colecciones:** Son conjuntos más específicos de documentación (por ejemplo: *Tesis doctorales, Libros etc*)
- **Por Fecha de publicación:** Listando el número de elementos que se dese por la fecha en el que fueron publicados originalmente, o por la fecha en el que se añadieron al repositorio, de forma ascendente o descendente.
- **Por Autor:** Se obtendrá un listado de todos los autores cuyas obras se encuentran disponibles.
- **Por Título de la obra:** Se listan todos los objetos disponibles, permitiendo listar los que comiencen por determinada letra o introducir los primeros caracteres del título para un acceso más eficaz.
- **Por Etiquetas:** Las etiquetas son palabras clave que definen la temática sobre la que trata un documento, por lo cuál pueden resultar de gran utilidad para listar solamente aquellos documentos que pertenezcan a la categoría que el usuario esté buscando (por ejemplo: para un experto en antropología le sería muy útil una etiqueta *Antropología*). Un mismo elemento puede disponer de varias etiquetas.

AI.3.3 Búsquedas

Se puede buscar un texto determinado, que se encuentre en cualquier campo de un documento (título, autor, fecha, texto contenido en el documento etc) mediante el buscador del panel derecho.

Por ejemplo, si queremos buscar obras cuyo autor se apellida “Camino” podemos utilizar dicho buscador:

A screenshot of a search bar interface. It consists of a white rectangular input field with a thin blue border. Inside the field, the word "Camino" is typed in a dark grey font. To the right of the input field is a green square button containing a white magnifying glass icon, representing the search function.

Figura 38. Buscador

No obstante, como es una palabra relativamente común la búsqueda arroja múltiples resultados no deseados se puede afinar la búsqueda por medio de la aplicación de *Filtros avanzados de búsqueda*, con los cuales se puede buscar por el valor de un determinado parámetro (Título,Autor,Etiquetas o Fecha de Publicación):



torio Nop ▼ Camino Ir

Mostrar filtros avanzados

Figura 39. Búsqueda avanzada

Para este ejemplo, de entre los filtros disponibles, como se pretende buscar un autor que contiene un apellido determinado, se escogerá el filtro por autor y buscar la palabra contenida “Camino”:



Use filtros para refinar sus resultados.

Autor ▼ Contiene ▼ Camino

Título

Autor

Etiquetas

Fecha

Mostrando ítems de 0 a 0

Figura 40. Filtros de búsqueda

Al utilizar el filtro (botón *Aplicar*), devolverá como resultado la obra buscada cuyo autor tiene el apellido Camino:



Mostrando ítems 1-1 de 1

La percepción de calidad, valor y satisfacción de padres y deportistas adultos

Camino López, Matías; García Fernández, Jerónimo (2014)

Figura 41. Resultado de búsqueda con filtros

AI.3.4 Registro de Usuario

Para tener acceso a más funcionalidades en el repositorio, será necesario crear una nueva cuenta mediante correo electrónico.

Para crear un usuario hay que acceder al *Login* de la esquina superior derecha:



Figura 42. Acceso a usuarios

En el apartado *Registrar un nuevo usuario*, se entra en el enlace *Pulse aquí para registrarse*. Se solicitará el correo electrónico con el que se accederá a la cuenta:

Una captura de pantalla de un formulario de registro. En la parte superior hay una barra de progreso con tres etapas: "Verifique el correo electrónico" (activa, en verde), "Crear perfil" (desactiva, en gris) y "Terminado" (desactiva, en gris). El texto principal dice: "Registrar una cuenta para suscribirse a colecciones para recibir por correo electrónico las modificaciones y las nuevas incorporaciones". Debajo, el campo "Dirección de correo electrónico: *" contiene el texto "sezabalam@alumnos.unex.es" en un campo de entrada con fondo amarillo. Un mensaje de ayuda indica: "Esta dirección será verificada y utilizada como su nombre de acceso." En la parte inferior izquierda hay un botón "Registro".

Figura 43. Registro de usuario (1)

Una vez introducido y siguiendo con el proceso de registro (Botón *Registro*) se notificará de que se ha enviado un correo a la dirección que se haya introducido para verificar la creación de la cuenta

Una captura de pantalla de un mensaje de correo electrónico. El asunto es "Correo de verificación enviado". En la parte superior hay una barra de progreso con tres etapas: "Verifique el correo electrónico" (activa, en verde), "Crear perfil" (desactiva, en gris) y "Terminado" (desactiva, en gris). El cuerpo del mensaje dice: "Se ha enviado un correo a sezabalam@alumnos.unex.es que contiene una URL especial e instrucciones de registro."

Figura 44. Registro de usuario (2)

En el buzón de correo se habrá recibido un nuevo mensaje que contendrá la URL a la que hay que acceder para completar el registro.

Por favor, abra el siguiente enlace para completar su registro como usuario en el Repositorio Nopoki:

<http://sight.unex.es/xmlui/register?token=ffe36f304f8ac65c7b7e49c344fe35ce>

Si necesita ayuda o soporte con su cuenta, por favor escriba al e-mail del administrador:
proyectonopoki@gmail.com
El administrador.

Figura 45.Registro de usuario (3)

Dicho enlace lleva al último paso, que es el de la creación del perfil, para lo cual hay que aportar una serie de datos personales y elegir una contraseña de acceso:

Verifique el correo electrónico → **Crear perfil** → Terminado

Identificar

Dirección de correo electrónico: sezabalam@alumnos.unex.es

Nombre: *

Apellido: *

Teléfono de contacto:

Idioma: Spanish

Seguridad

Por favor, introduzca una clave en la casilla superior. Confírmela volviendo a teclearla en la segunda casilla. Debería tener al menos seis caracteres.

Contraseña: *

Repita para confirmar: *

Figura 46.Registro de usuario (4)

Tras introducir dichos datos, se procederá a dar por terminado el proceso de registro de usuario (Botón *Completar registro*), y se mostrará un mensaje de confirmación de que todo ha ido correctamente.



Figura 47.Registro de usuario (5)

AI.3.5 Petición de Publicación

Como inicialmente el usuario creado no tendrá permisos de publicación de archivos sobre ninguna de las colecciones, deberá pedirse expresamente al administrador del repositorio sobre qué colección o colecciones se desea introducir nuevo contenido.

Para ello, existe un formulario en el cual indicar dicha solicitud, en la esquina superior izquierda en el campo *Solicitudes*:

En dicho formulario, se deberá indicar el correo electrónico de la cuenta y en el cuadro de texto enumerar aquellas colecciones en las que se desea contribuir, así como si se desean crear nuevas colecciones o comunidades.

Pongamos el ejemplo de solicitar permiso para la publicación en la colección “*Colección 1*” de la comunidad “*Pruebas*”:

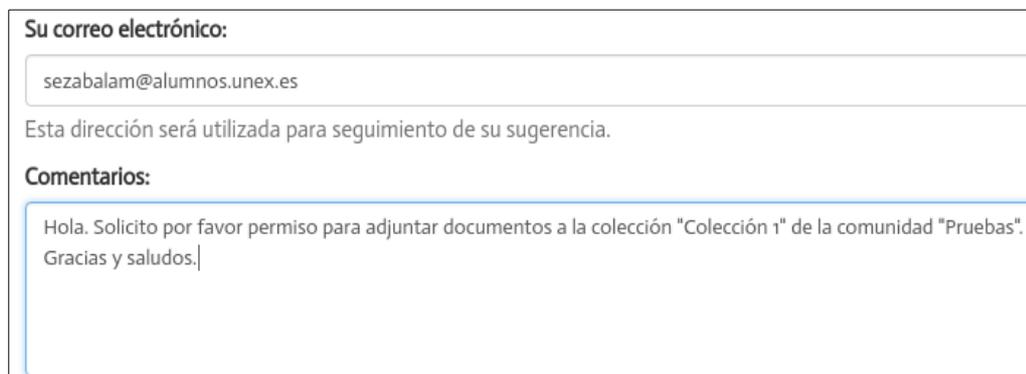
The image shows a form titled "Su correo electrónico:" with a text input field containing "sezabalam@alumnos.unex.es". Below this, it says "Esta dirección será utilizada para seguimiento de su sugerencia." The next section is "Comentarios:" with a text area containing the text: "Hola. Solicito por favor permiso para adjuntar documentos a la colección "Colección 1" de la comunidad "Pruebas". Gracias y saludos." The form has a blue border and a light blue background.

Figura 48.Formulario de petición

Una vez confirmado el envío, el usuario deberá esperar a la confirmación del administrador, que será el que decida si se le conceden o no los privilegios, para poder comenzar con la subida de archivos.



Figura 49. Correo de confirmación del Administrador

AI.3.6 Publicación de Archivos

Si se ha recibido el visto bueno por parte del Usuario Administrador del repositorio, se puede proceder a enviar elementos a aquellas colecciones en las que se tengan privilegios (En este ejemplo, la colección “Colección 1”)

Se accede mediante el explorador de *Comunidades & Colecciones* a la colección elegida,

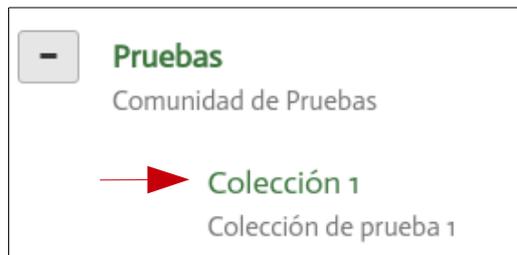


Figura 50. Seleccionar colección

Dentro de esta colección, existirá un campo para el envío:

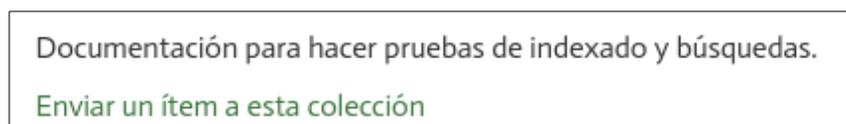


Figura 51. Publicar documento (1)

También es posible desde la opción *Envíos* del panel lateral en la sección *Mi Cuenta*:



Figura 52. Publicar documento (2)

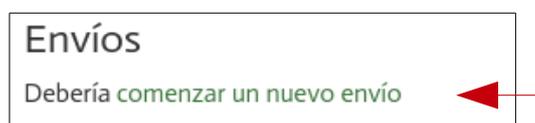


Figura 53. Publicar documento (3)

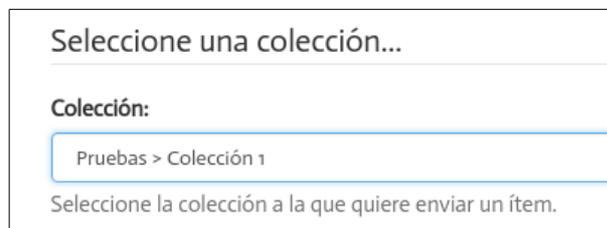


Figura 54. Publicar documento (4)

Formulario de publicación

Para el proceso de subida de ficheros, es necesario completar un formulario de envío donde se introduce la información correspondiente al documento en cuestión. Los campos que aparecen con asterisco son de obligada inserción.

Describir el ítem

Autor/es:
Apellido, p.ej. Pérez Nombre(s), p.ej. Manuel/

Introduzca nombre y apellidos del autor (o autores si hay varios).

Título: *

Introduzca el título principal del artículo.

Otros Títulos:

Si existe otro título, introdúzcalo

Fecha de publicación: *

Año Mes Día

Introduzca la fecha de publicación del elemento (al menos el año).

Editor:

Introduzca el editor encargado de publicar este artículo..

Identificadores:

ISSN

Si el artículo lleva asociado algún código de identificación, por favor, introdúzcalo.

Tipo:

Animación

Figura 55. Publicar documento (5)

Las dos primeras páginas son de descripción en las que se deben aportar campos como el título de la obra, el autor o autores, la fecha original de publicación, el tipo de documento, el idioma (si aplica), las etiquetas que hagan referencia al contenido o un breve resumen de lo que contiene, entre otros parámetros.

Tras esta fase de describir, será necesario adjuntar el archivo o archivos (si son varias partes) que serán almacenados en el repositorio:



The screenshot shows a web interface for uploading a document. At the top, there is a horizontal navigation bar with six buttons: 'Describir', 'Describir', 'Subir', 'Revisar', 'Licencia', and 'Completar'. Below this, the main heading is 'Subir fichero(s)'. Underneath, there is a section labeled 'Fichero: *' with a button 'Seleccionar archivo' and the text 'Ningún archivo seleccionado'. A paragraph of instructions follows: 'Por favor, introduzca la ruta completa del fichero en su ordenador que corresponda con el ítem. Si pincha en "Examinar...", se abrirá una ventana que le permitirá seleccionar un fichero de su ordenador.' Below this is a section labeled 'Descripción del fichero:' with a large text input field. A note below the field says: 'Si lo desea, puede describir brevemente el contenido de este fichero, por ejemplo "Artículo principal", o "Lectura de los datos del documento".' At the bottom left, there is a button labeled 'Subir fichero y añadir otro más'.

Figura 56.Publicar documento (6)

Una vez elegido dicho archivo, se pueden revisar los datos que se han introducido para corregir alguno de ellos antes de completar el envío.

En caso de confirmar que la información es correcta, queda el último paso, que es la concesión de la licencia de reproducción, traducción y distribución global de la obra. Es necesario aceptar dicha concesión de licencia para poder realizar la publicación:

Licencia de distribución:

Conceder licencia

Figura 57.Publicar documento (7)

Todos los pasos han sido completados, y se nos notifica de ello, dándonos la opción de realizar otro envío o ver los envíos realizados.



Figura 58. Publicar documento (8)

Se puede comprobar desde la página principal del *Repositorio Nopoki* que se ha suscrito correctamente en la lista de los últimos elementos añadidos:



Figura 59. Publicar documento (9)

AI.3.7 Visualización de estadísticas

En el apartado *Estadísticas* → *Ver Estadísticas de Uso*, se podrá observar el listado de los documentos más accedidos:

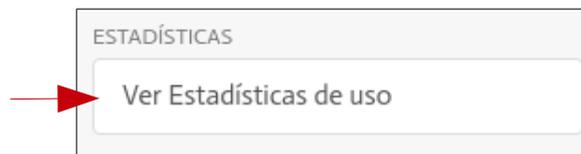


Figura 60. Ver estadísticas (1)

Estadísticas	
Número total de visitas	
	Visualizaciones
Curriculum Vitae	25
[Reseña] Ballesta Pagán, J. (2016). Acuse de Recibo: entre la realidad y el deseo en educación	21
Documento de prueba	13
Título de Ejemplo	12
Relatos Matsigenkas	10
Río Amazonas	6

Figura 61. Ver estadísticas (2)

ANEXO II - Guía del Administrador

Este documento va destinado a la persona o grupo que haya sido designado como Administrador del *Repositorio Nopoki*.

Su papel será el de gestionar todos los recursos que existan en el repositorio, creando para ello las comunidades y colecciones que estime oportuno, siendo el único actor que puede crear colecciones.

Será también responsable de la gestión de usuarios y privilegios, decidiendo a qué recursos pueden tener acceso determinados usuarios, permitiendo publicación de contenido a ciertas colecciones a los grupo de usuarios que lo hayan solicitado, así como posibilitar la edición de comunidades y colecciones.

Llevará un control exhaustivo de las tareas que se han realizado mediante estadísticas, así como otras tareas exclusivas como la importación de elementos a través de metadatos o ficheros comprimidos, tareas de curación etc.

All.1 Creación de Comunidades y Colecciones

- Para crear una nueva **Comunidad**:
 - Panel Derecho → Contexto → Crear Comunidad
 - Introducir datos descriptivos:
 - Nombre
 - Descripciones
 - Textos introductorios y de copyright
 - Novedades
 - Logotipo



Figura 62.Crear comunidad (1)

Editar metadatos de una nueva comunidad de primer nivel

Nombre:

Descripción breve:

Texto introductorio (HTML):

Texto de copyright (HTML):

Novedades (HTML):

Figura 63. Crear Comunidad (2)

- Botón Crear para confirmar
- Si se desea nombrar a un usuario o grupo como administrador de la comunidad, para que tengan permisos de creación de subcomunidades o colecciones y capacidad de asignar permisos sobre ellas, así como de publicar datos, se debe realizar en la pestaña Asignar roles (por defecto se procurará no hacer desde un inicio para ninguna comunidad salvo petición expresa por parte de algún usuario).
- Una vez creada, se puede ver listada en la sección Comunidades & Colecciones:

Elija una comunidad para listar sus colecciones

[Comunidad de Sergio](#)
Comunidad creada para el manual del administrador del Repositorio.

Figura 64. Crear comunidad (3)

- Para crear una nueva **Colección**:
 - Acceder a la Comunidad elegida (en este caso, la recién creada Comunidad de Sergio)
 - Panel Derecho → Contexto → Crear Colección
 - El formato del formulario de creación es idéntico al de las Comunidades, con dos campos más
 - Licencia
 - Origen



Figura 65. Crear colección (1)

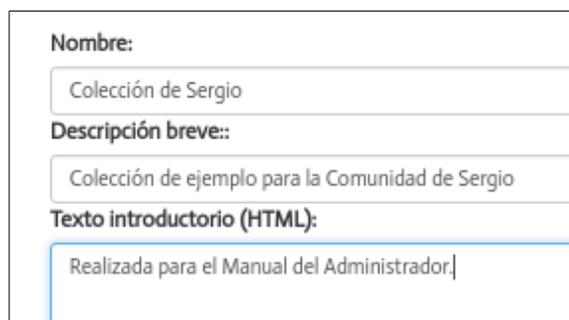
A screenshot of a form for creating a collection. It has three sections: 'Nombre:' with the value 'Colección de Sergio'; 'Descripción breve:.' with the value 'Colección de ejemplo para la Comunidad de Sergio'; and 'Texto introductorio (HTML):' with the value 'Realizada para el Manual del Administrador,'.

Figura 66. Crear colección (2)

- Una vez introducidos los datos, confirmar mediante el botón Crear
- Cuando se haya creado, se procederá a la asignación de roles. Al igual que sucedía con las comunidades, por defecto no se asignará permisos a ningún usuario excepto que la creación de dicha colección se haya

realizado a petición expresa de un usuario. La gestión de los roles se detallará en secciones siguientes.

- También se permite indicar el origen de los datos (Pestaña Origen del Contenido) del repositorio. Como los datos serán aportados por los usuarios y no está habilitado el servicio de recolección OAI para tomar ficheros de fuentes externas, la opción marcada deberá ser siempre:

Origen del Contenido:

- Esta es una colección DSpace estándar

Figura 67. Crear colección (3)

All.2 Gestión de Usuarios

El administrador tiene el control sobre todos los usuarios del repositorio, en el apartado del panel derecho *Administrativo* → *Control de acceso* → *Personas*

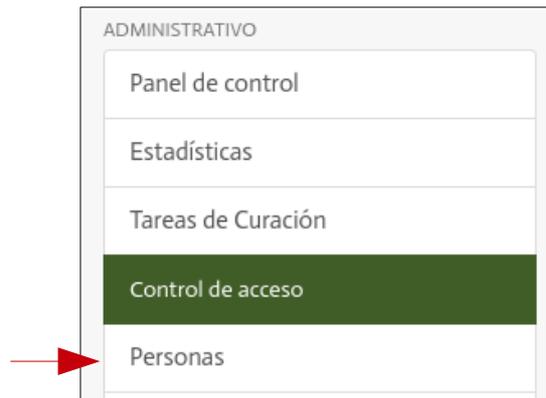


Figura 68. Administrar usuarios

Las opciones disponibles para con los usuarios son:

- **Listar Usuarios**
- **Creación de Usuarios**
 - Acciones → Crear un nuevo usuario



Figura 69. Crear usuario

- **Eliminación de Usuarios**

- Seleccionar en la lista el/los usuario/s deseado/s (salvo aquellos que posean permisos de administración) y botón *Borrar Usuarios*

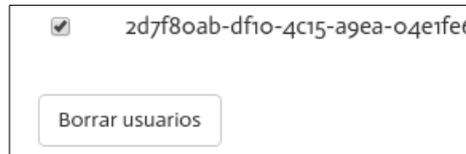


Figura 70. Eliminar usuario

All.2.1 Gestión de los grupos

Es posible crear nuevos grupos con los usuarios que se deseen. Esto es útil a la hora de asignar permisos de publicación o modificación sobre una determinada comunidad o colección para un conjunto de personas específico.

Por ejemplo, si existiese una comunidad para la *Universidad de Extremadura*, podría ser interesante crear un grupo *Universidad de Extremadura* en el que formasen parte todos los usuarios registrados con el dominio *@unex.es*, para posteriormente otorgarle ciertos permisos sobre dicha comunidad.

El apartado de administración de los grupos se encuentra en la sección del panel derecho *Administrativo* → *Control de Acceso* → *Grupos*

Las opciones que ofrece dicha sección son:



Figura 71. Administrar grupos

- **Listar grupos**
- **Crear nuevo grupo**
 - Acciones → Crear un nuevo grupo
 - Introducir un nombre
 - Añadir miembros
 - Usuarios → Seleccionar usuario a integrar y opción *Añadir*
 - Otros grupos

En este caso se creará un grupo de prueba con dos usuarios de ejemplo:

Cambiar el nombre del grupo:

Buscar miembros para añadir:

ID	Nombre	Correo electrónico	
ca125022-d71f-482a-9cd8-baa1da3f5240	Sergio Zabala	moriquillo@gmail.com	<input type="button" value="Añadir"/>
78f1223a-ebb8-4399-b9db-fe557ee18ad1	Sergio Zabala Mesonero	proyectonopoki@gmail.com	<input type="button" value="Añadir"/>
zd7f80ab-df10-4c15-a9ea-04e1fe61509b	Sergio Zabala Mesonero	szabala@animbos.com	Pendiente
bbfo4216-4b4e-49cb-ad63-30db6d65c701	Sergio Zabala Mesonero	sezabalam@alumnos.unex.es	Pendiente

Figura 72. Crear grupo (1)

El último paso será el de confirmar la creación mediante el botón *Guardar*.

El grupo una vez creado aparecerá listado, y por defecto no tendrá ninguna comunidad o colección asociadas.

ID	Nombre	Miembros	Comunidad / Colección
<input type="checkbox"/> 60c24doa-8918-49f3-9003-d25a85cccodd	Grupo de Prueba	2	

Figura 73. Crear grupo (2)

- **Borrar grupos**
 - Seleccionar aquellos grupos que se deseen eliminar y botón *Borrar grupos* y confirmar borrado.

<input checked="" type="checkbox"/>	61e54790-1253-4df2-a6fc-a31559f70a2e	Crear Colecciones	1
<input checked="" type="checkbox"/>	f91a4f15-4odf-4be7-8ace-77325bb3f7do	Publicadores	2
<input checked="" type="checkbox"/>	e9695679-b7fa-4336-9cc3-4535779od8cb	Pruebas - Colección 1	1

Borrar grupos

Figura 74. Eliminar grupo (2)

All.3 Gestión de Permisos

Uno de los aspectos más relevantes para el usuario Administrador es el de la asignación de permisos sobre los distintos recursos del repositorio digital.

Así pues, el administrador tendrá la posibilidad de añadir un nuevo privilegio para un grupo de usuarios previamente creado sobre una comunidad o colección. Dichos privilegios pueden ser de distintos tipos:

- **Lectura (READ)** : Por defecto la tendrán todas las comunidades y colecciones para todos los usuarios (incluidos visitantes sin registrar)
- **Modificación (WRITE)**
- **Agregación (ADD)**: Será el privilegio necesario para permitir la subida de contenido a un grupo de usuarios registrados.
- **Eliminación (REMOVE)**
- **Administración (ADMIN)**: Con este privilegio se otorga a un grupo la capacidad de gestionar todo el contenido de una comunidad/colección, así como la gestión de los roles de ésta.

All.3.1 Asignación de privilegios

Se va a mostrar con un ejemplo la secuencia de pasos para asignar privilegios sobre una colección a un grupo específico de usuarios registrados.

El ejemplo propuesto será el de permitir la publicación a un grupo sobre la colección de prueba anteriormente creada. Se considera de utilidad explicar este proceso, ya que será repetido en múltiples ocasiones por el administrador.

1. Acceder al panel de gestión de permisos: *Administrativo* → *Control de acceso* → *Autorizaciones*



Figura 75. Gestionar permisos

2. Seleccionar la colección a la que dar permisos de publicación:



Figura 76. Acceso a colección

3. Añadir un nuevo privilegio:



Figura 77. Añadir privilegio (1)

4. Otorgarle un nombre y seleccionar los permisos pertinentes (en este caso la autorización *ADD*, relativa a la publicación de contenido):

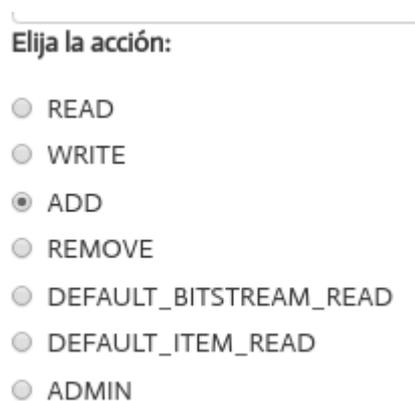


Figura 78. Añadir privilegio (3)

5. Elegir el grupo, en este caso el de pruebas creado anteriormente:



The image shows a web interface element for selecting a group. It consists of a rectangular box with a light gray border. At the top left of the box, the text "Elija un grupo:" is displayed in a bold, dark font. Below this text is a horizontal dropdown menu. The menu is currently open, showing a single option: "Grupo de Prueba". The text "Grupo de Prueba" is centered within the dropdown menu.

Figura 79. Añadir privilegio (4)

6. Confirmar los cambios (Botón *Guarda*)

Este es el proceso seguido para asignar permisos de publicación a usuarios, y se hará de igual forma para comunidades.

Por defecto, ninguna comunidad o colección contará con dicha autorización, y será necesario recibir por correo electrónico la petición expresa por parte del usuario **registrado** de poder añadir documentos y a qué comunidad/colección.

Esto lo realizará el usuario mediante la sección de *Solicitudes*, tras lo cuál se recibirá un mensaje en la cuenta de correo del administrador con el siguiente formato:



Figura 80. Correo de petición de permiso de publicación

Tras lo cual , será el administrador el que decida acceder a permitirle lo que demanda, siguiendo el procedimiento anterior.

All.4 Otras funcionalidades

Además de las arriba enumeradas, un usuario Administrador del repositorio podrá realizar otras tareas exclusivas que pueden ser de mucha utilidad:

- **Exportaciones:**
 - Exportar Colecciones y Comunidades en formato .zip , que se enviará al correo, así como sus metadatos (formato *CSV*).
 - Exportar elementos en formato .zip que podrá ser descargado desde la cuenta de correo, o sus metadatos (en *CSV*)
- **Importaciones**
 - Importación de comunidades,colecciones o elementos a partir de sus metadatos (en *CSV*)
 - Importación de elementos en .zip
- **Control de las tareas que se han realizado:** N.º de archivos almacenados, n.º de accesos de usuarios, búsquedas realizadas etc.
- **Visualización de estadísticas:** Términos buscados y cuántas veces, artículos más visualizados etc.

ANEXO III - Guía del Desarrollador

Esta guía pretende reflejar los pasos necesarios para implantar el repositorio digital en un entorno de similares características al utilizado en el proyecto.

Está enfocado a una posible futura migración a otro soporte y de relevo del administrador del sistema, cuyo papel recae en la actualidad en el creador de este proyecto.

Se enumeran los pasos seguidos de una forma sencilla y directa, sin recalcar en detalles, para facilitarle el proceso al administrador del sistema encargado de realizar el posible despliegue.

AIII.1 Requisitos

- Máquina (servidor físico, virtualizado u ordenador personal) con un mínimo de 8 GB de RAM, 256 GB de almacenamiento.
- Sistema Operativo basado en Debian (p.ej: *Ubuntu 16.04*, *Linux Mint 17*, *Debian 8 etc*)
- Recomendable: Sistema de almacenamiento externo (*SAN*, *NFS*, *SMB*, *AWS S3*, *Nextcloud etc*)

AIII.2 Instalación de la plataforma

AIII.2.1 Instalación de Apache Tomcat y sus dependencias

Estos son los pasos a seguir para instalar el servidor web:

- **Abrir terminal e insertar los siguientes comandos:**

```
#Instalar Java 7
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update
sudo apt-get install openjdk-7-jdk
# Configurar Variable de Entorno de Java
sudo vim ~/.bashrc
# Insertar la línea export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64

#Instalar Apache Ant
sudo apt-get install ant

#Instalar Maven
sudo apt-get install maven

#Instalar Tomcat 7 (desde fuentes externas) – Acceder a usuario root
sudo su -
cd /tmp/
curl -O
http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.70/bin/apache-tomcat-7.0.70.tar.
gz

tar xzvf apache-tomcat-7*tar.gz -C /opt/tomcat --strip-components=1

#Añadir usuario 'tomcat' y darle permisos sobre el directorio de instalación
groupadd tomcat
useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat

cd /opt/tomcat/
chgrp -R tomcat conf
chmod g+rx conf
chmod g+r conf/*
chown -R tomcat work/ temp/ logs/
```

- Crear un nuevo servicio para Apache Tomcat y cargarlo en el `systemd`:
 - Se crea un nuevo fichero `/etc/systemd/system/tomcat.service` con el contenido:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xmx2048M -Xms2048M
-XX:PermSize=512m -XX:MaxPermSize=512m -XX:NewSize=256m -server
-XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true
-Djava.security.egd=file:/dev/./urandom'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

- Generar un nuevo servicio:

```
systemctl daemon-reload
systemctl start tomcat
systemctl stop tomcat
systemctl enable tomcat
```

AIII.2.2 Instalación de Dspace y sus dependencias

```
#Añadir usuario dspace a Ubuntu
useradd -m dspace
sudo su - dspace
# Instalar PostgreSQL 9.5
sudo apt-get update
sudo apt-get install postgresql-9.5
#Crear usuario dspace y base de datos dspace en postgresql
sudo su - postgres
createuser --username=postgres --no-superuser --pwprompt dspace
createdb --username=postgres --owner=dspace --encoding=UNICODE dspace
psql --username=postgres dspace -c "CREATE EXTENSION pgcrypto;"

# Crear directorio que alojará el código fuente descargado
sudo mkdir /build
sudo chmod -R 777 /build
cd /build
#Obtener código fuente y descomprimirlo en la carpeta de instalación
sudo wget
https://github.com/DSpace/DSpace/releases/tag/dspace-6.2/dspace-6.2-src-release.t
ar.gz

sudo tar -zxf dspace-6.2-src-release.tar.gz

cd /dspace-6.2-src-release
#Compilar con Maven
sudo mvn -U package
#Instalar con Ant
cd dspace/target/dspace-installer
sudo ant fresh_install

#Cambiar propiedad del directorio al usuario tomcat
sudo chown -R tomcat:tomcat /dspace
```

- **Configurar PostgreSQL**

- Editar el fichero `/etc/postgresql/9.5/main/pg_hba.conf` con permisos root y añadir la siguiente línea:

```
local all dspace md5
```

- Reiniciar servicio: `sudo service postgresql restart`

- **Configurar Webapps incluidas en Tomcat**

- Por cada aplicación, por ejemplo *xmlui*, añadir un fichero .xml con el nombre al directorio `/opt/tomcat/conf/Catalina/localhost/`

```
sudo touch /opt/tomcat/conf/Catalina/localhost/repositorio.xml
```

- El contenido de estos fichero tendrá este formato:

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

- Reiniciar *Tomcat*:

```
sudo systemctl restart tomcat.service
```

- **Creación de un usuario administrador:**

```
sudo /dspace/bin/dspace create-administrator
```

Este es el proceso básico para la instalación de un CMS *Dspace* en un entorno Ubuntu, y será necesario como base a la implantación de la plataforma de repositorio digital *Nopoki*.

AIII.3 Migración del Repositorio

Como el soporte donde se encuentra ahora desplegada toda la infraestructura de la plataforma del *Repositorio Nopoki* está cedida por el Servicio de la Escuela Politécnica (iEPCC) de manera temporal, será necesaria una migración a un nuevo sistema de forma definitiva.

Por este motivo, el nuevo personal encargado de la administración del sitio web, deberá solicitar al actual los siguientes recursos para poder realizar el traslado:

- Copias de seguridad de la Base de Datos
- Copia del Directorio de Ficheros (*assetstore*)
- Ficheros de Configuración de DSpace
- Ficheros de Configuración del servidor Tomcat
- *Scripts* para la automatización de algunas tareas
- Ficheros de configuración de Ubuntu

Así pues., una vez acordado el destinatario del producto, éste deberá ponerse en contacto con el desarrollador del proyecto para proporcionarle los elementos arriba citados, además de realizar un seguimiento detallado de los pasos a seguir para dicha migración, así como el asesoramiento en la resolución de posibles problemas que surjan.

Para realizar este procedimiento, deberá ponerse en contacto con la cuenta de correo electrónico proyectonopoki@gmail.com