

Original Research / Artículo Original - Tipo 1

R. B. Repository: Reference bibliographies repository for Drupal 7

Fabián Ortiz Collazos / fabianortiz@unicauca.edu.co

Beatriz Elena Hurtado Hurtado / beatrizhurtado@unicauca.edu.co

Eduardo Rojas Pineda / erojas@unicauca.edu.co

Universidad del Cauca, Popayán-Colombia

ABSTRACT This article, a result of a research project, the characteristics and functions of a module developed for managing bibliographic references about the content management system Drupal Version 7 are exposed. It integrates several functions that allows to host the information in a repository, of both bibliographic and users, and includes a function based on the Scopus API to obtain in a semiautomatically way metadata from publications, which simplifies and streamlines the information introduction and extraction. Acquiring metadata from publications would have a semiautomatic component which would reduce the amount of information that should be entered manually.

KEYWORDS Bibliographic repository; semiautomatic capture of information; Scopus API; module; Drupal.

R. B. Repository: repositorio de referencias bibliográficas para Drupal 7

RESUMEN En este artículo, resultado de un proyecto de investigación, se exponen las características y funciones de un módulo desarrollado para la gestión de referencias bibliográficas sobre el sistema gestor de contenidos Drupal, versión 7. Integra varias funciones que permiten albergar en un repositorio la información, tanto bibliográfica, como de usuarios, e incluye una función basada en el API de Scopus para obtener de manera semiautomática los metadatos de las publicaciones, con lo que simplifica y hace más eficiente la introducción y extracción de información. la adquisición de los metadatos de publicaciones tendría un componente semiautomático lo que reduciría la cantidad de información que se debería introducir manualmente.

PALABRAS CLAVE Repositorio bibliográfico; captura semiautomática de información; API de Scopus; módulo; Drupal.

R. B. Repository: repositório de referências bibliográficas para Drupal 7

RESUMO Neste artigo, resultado de um projeto de pesquisa, são expostas as características e funções de um módulo desenvolvido para a gestão de referências bibliográficas sobre o sistema de gerenciamento de conteúdo Drupal, versão 7. Integra várias características que permitem armazenar num repositório informação bibliográfica e de usuários, e inclui uma função com base na Scopus API para obter de forma semiautomática os metadados das publicações, o que simplifica e faz mais eficiente a introdução e a extração de informação. A aquisição dos metadados de publicações teria um componente semiautomático, o que reduziria a quantidade de informação que deveria ser introduzida manualmente.

PALAVRAS-CHAVE Repositório bibliográfico; captura semiautomática de informação; Scopus API; módulo Drupal.

I. Introduction

Repositories, also called deposits or digital files, emerge as one of the key strategies of the open access movement, so the scientific community and public in general can have free and open access to all types of investigative content.

A repository can be understood as a centralization and distribution system of information that is much more organized than conventional databases; generally speaking, it can host digital files or metadata intellectual products of scientific and academic nature (Flores & Sánchez, 2007).

The Department of Telematics of Faculty of Electronics and Telecommunications Engineering [FIET] of the University of Cauca has a website which has information about publications made by teachers, students or research groups associated with programs of Master and PhD in Telematics Engineering, designed on the Drupal content management system. Due to the way this information is stored and consulted, modifications can be very wasteful and inefficient, a factor that can be improved if these data are stored in a repository designed in a way that inserting or removing information can be easier.

The project consists of develop a module to function as a repository for managing bibliographic references about CMS [Content Management System] Drupal 7, capable of capturing and storing, semiautomatically, metadata from article type publications using resources offered by the Scopus API.

II. Method

The construction of the bibliographic repository consists of three phases: the first, “Bibliographic analysis”, it is used the model for documentary research set out by Serrano Castaño (2005); the second, “Design and implementation of bibliographic repository”, it comprises the functional analysis of the system and the iterative and incremental software development; in the third and final phase, “Evidence”, integration and proper functioning of the developed system is verified.

Development environment

Once the repository structured was set out, after the bibliographic analysis and considering the tools that are compatible with Drupal v7.43, were defined those that were going to be used during the project development.

The first step was to define which operating system to work with; it was decided to use Ubuntu 14.04 LTS, because it is free and is widely used as a server.

I. Introducción

Los repositorios, también denominados depósitos o archivos digitales, surgen como una de las estrategias fundamentales del movimiento de acceso abierto, para que la comunidad científica y el público en general tengan acceso gratuito y libre a todo tipo de contenido investigativo.

Un repositorio se puede entender como un sistema de centralización y distribución de la información mucho más organizado que las bases de datos convencionales; generalmente puede albergar los archivos digitales y/o los metadatos de los productos intelectuales de carácter científico y académico (Flores Cuesta & Sánchez Tarragó, 2007).

El Departamento de Telemática de la Facultad de Ingeniería Electrónica y Telecomunicaciones [FIET] de la Universidad del Cauca cuenta con un portal en el que se tiene información sobre las publicaciones realizadas por docentes, estudiantes y/o grupos de investigación asociados a los programas de Maestría y Doctorado en Ingeniería Telemática, diseñado sobre el sistema gestor de contenidos Drupal. Debido a la forma en que se almacena y se consulta esta información, realizar modificaciones puede llegar a ser muy dispendioso e ineficiente, factor que se puede mejorar si estos datos se almacenan en un repositorio diseñado de forma tal que introducir o extraer información sea más sencillo.

El proyecto consiste en desarrollar un módulo que funcione como un repositorio para la gestión de referencias bibliográficas sobre el CMS [Content Management System] Drupal 7, capaz de capturar y almacenar, semiautomáticamente, los metadatos de las publicaciones tipo artículo empleando los recursos ofrecidos por la API de Scopus.

II. Método

La construcción del repositorio bibliográfico consta de tres fases: en la primera, “Análisis bibliográfico”, se emplea el modelo para la investigación documental planteado por Serrano Castaño (2005); la segunda, “Diseño e implementación del repositorio bibliográfico”, comprende el análisis funcional del sistema y el desarrollo iterativo e incremental del software; en la tercera y última, “Pruebas”, se verifica la integración y el correcto funcionamiento del sistema desarrollado.

Entorno de desarrollo

Una vez se planteó la estructura del repositorio, tras el análisis bibliográfico y considerando las herramientas compatibles con Drupal v7.43, se definieron aquellas a utilizar durante el desarrollo del proyecto.

El primer paso fue definir con qué sistema operativo trabajar; se decidió emplear Ubuntu 14.04 LTS, por ser libre y por el hecho de ser muy utilizado como servidor.

Para la correcta instalación de Drupal es indispensable contar con un servidor web, un gestor de base de datos y un lenguaje de programación, la comunidad Drupal recomienda Apache, MySQL y PHP, respectivamente, de este último

se utilizó la versión 5.6. Para facilitar el manejo y confirmar la efectividad de las modificaciones que se realizan sobre la base de datos, se optó por la herramienta de gestión phpMyAdmin.

La herramienta elegida para la edición de texto fue Sublime Text 3, debido a la amplia gama de características que presenta, muchas de las cuales apoyan y facilitan el trabajo del desarrollador.

III. Trabajos relacionados

En la primera fase de la metodología planteada se encontró una gran variedad de repositorios públicos y privados. No se halló una clasificación estándar, pero se pudo deducir la idea de que un repositorio de referencias bibliográficas alberga información de índole académica e investigativa principalmente.

Los repositorios de referencias bibliográficas son muy poco comunes, según la información recopilada durante la fase de análisis bibliográfico, pero se encontraron documentos que afirmaban la existencia de algunos. Estas publicaciones exponen repositorios que presentan similitudes con el trabajo a realizar, entre los más representativos se tienen:

- Repositorio CS OER: portal diseñado con el fin de facilitar la búsqueda de recursos educativos y la navegación por varios repositorios y sus contenidos (Dicheva & Dichev, 2014).
- BoW: Feitelson (2000) describe el diseño de un repositorio basado en una definición de repositorio bibliográfico, que facilite la búsqueda de documentación bajo conceptos deseados; en el documento se clasifica esta herramienta bibliográfica como un DyRI [*Dynamic Ranked Information-space*].
- Metabuscador: Chacón y Cruz (2012) presentan un metabuscador para referencias bibliográficas que funciona en torno a una biblioteca y revistas en Costa Rica y repositorios externos.
- ISBIB: Chua, Lan, Cousins, Mohan y Straub (2002) lo describen con funcionalidades similares a las del proyecto, con un ingreso manual de información por parte del administrador.
- OER Commons: repositorio bibliográfico que ofrece material educativo gratuito para docentes, estudiantes, aprendices y la comunidad académica en general (ISKME, 2007).

Módulos relacionados

Debido a que el proyecto pretende desarrollar un módulo para Drupal 7, una vez definidas las funciones que debe desempeñar, es fácil identificar aquellos módulos existentes que pueden servir como base o apoyo para el desarrollo planteado en este artículo. Los módulos que se mencionan a continuación, fueron desarrollados y publicados por el grupo Biblioteca de Drupal.

- Bibliography Module (2006): permite gestionar y mostrar publicaciones además de importar y exportar con diferentes estilos y formatos.

For proper installation of Drupal it is essential to have a web server, a database manager and a programming language, Drupal community recommends Apache, MySQL and PHP, respectively, for the last one it was used the 5.6 version. To facilitate management and confirm the effectiveness of the modifications that are performed on the database, it was chosen the phpMyAdmin management tool.

The tool of choice for editing text was Sublime Text 3, due to the wide range of characteristics that presents, many of which support and facilitate the work of the developer.

III. Related works

In the first phase of the proposed methodology it was found a big variety of public and private repositories. A standard classification was not found, but it can be deduced the idea that a repository of bibliographic references hosts information mainly of academic and research nature.

Repositories of bibliographic references are rare, according to information gathered during the bibliographic analysis phase, but documents asserting the existence of some of them were found. These publications expose repositories that have similarities with the work to be performed, the most representative include the following:

- CS OER Repository: Website designed to facilitate the search for educational resources and navigating multiple repositories and their contents (Dicheva & Dichev, 2014).
- BoW: Feitelson (2000) describes the design of a repository based on a definition of bibliographic repository that facilitates the search for documentation under desired concepts; in the document, this bibliographic tool is classified as a DyRI [*Dynamic Ranked Information-space*].
- Metasearch engine: Chacón y Cruz (2012) present a metasearch engine for bibliographic references that works around a library and journals in Costa Rica and external repositories.
- ISBIB: Chua, Lan, Cousins, Mohan and Straub (2002) describe it with similar features to the project, with manual data entry by the administrator.
- OER Commons: Bibliographic repository that provides free educational materials for teachers, students, trainees and academic community in general (ISKME, 2007).

Related modules

Because the project intends to develop a module for Drupal 7, once the functions to be performed are defined, it is easy to identify those existing modules that can serve as a base or support for the development proposed in this article. The modules mentioned below were developed and published by the Library group of Drupal.

- Bibliography Module (2006): It enables to manage and display publications as well as to import and export with different styles and formats.
- Book Post (2008): It uses the ISBN to provide the most relevant information about the book, such as the title, author and publisher.
- eXtensible Catalog Drupal Toolkit (2010): Modular system that offers several features for metadata management.
- Masterkey: Drupal (MkDru) (2014): It provides an interface for searching data through information standards that are compatible with platform PazPar2.
- OAI-PMH (2006): It provides an interface for protocol metadata harvesting OAI-PMH for the Bibliography module.
- Z39.50 Search (2006): It implements the Z39.50 protocol to retrieve information using other Drupal tools.

In addition to the modules developed by the Library group, there are other that turn around important functions that must be performed by the system on development, including:

- Taxonomy Manager (2007): It enables the management of taxonomies, individual or mass, because it works as a dynamic tree, besides having its own interface.
- Import/Export (2009): It facilitates exporting and importing information with extension “.cck” by site administrators.
- ONKI (2012): Module developed primarily for the Finns, has ontologies, thesauri and international vocabulary, which allows to tag information in an efficiently way.
- Easy import (2011): Module designed to easily import, as its name says, CSV and XML files type.
- Simple CSV Importer (2015): Very powerful feature that imports in CSV files almost any field, including text, image and taxonomy, among others.

- Book Post (2008): utiliza el ISBN para proporcionar la información más relevante sobre el libro, como el título, autor y *publisher*.
- eXtensible Catalog Drupal Toolkit (2010): sistema modular que dispone de varias funcionalidades para la gestión de los metadatos.
- Masterkey: Drupal (MkDru) (2014): ofrece una interfaz para la búsqueda de datos a través de estándares de información compatibles con la plataforma de PazPar2.
- OAI-PMH (2006): brinda una interfaz para el protocolo de recolección de metadatos OAI-PMH al módulo de Bibliografía.
- Z39.50 Search (2006): implementa el protocolo Z39.50 para recuperar información empleando otras herramientas de Drupal.

Además de los módulos desarrollados por el grupo Biblioteca, también se tienen otros que giran en torno a funciones importantes que debe desempeñar el sistema a implementar, entre ellos:

- Taxonomy Manager (2007): permite la gestión de taxonomías, individuales o en masa, gracias a que funciona como un árbol dinámico, además de contar con su propia interfaz.
- Import/Export (2009): facilita exportar e importar información con extensión “.cck” por parte de los administradores del sitio.
- ONKI (2012): módulo desarrollado principalmente para los finlandeses, cuenta con ontologías, tesauros y vocabulario internacional, lo que permite etiquetar la información de forma eficiente.
- Easy import (2011): módulo diseñado para importar fácilmente, como su nombre lo dice, archivos de tipo CSV y XML.
- Simple CSV Importer (2015): función muy potente que importa en archivos CSV casi cualquier campo, incluidos: texto, imagen y taxonomía, entre otros.

IV. Diseño e implementación

Para el diseño del repositorio se tuvo en cuenta la información de los trabajos relacionados recolectada durante el proceso de documentación. Los módulos relacionados se

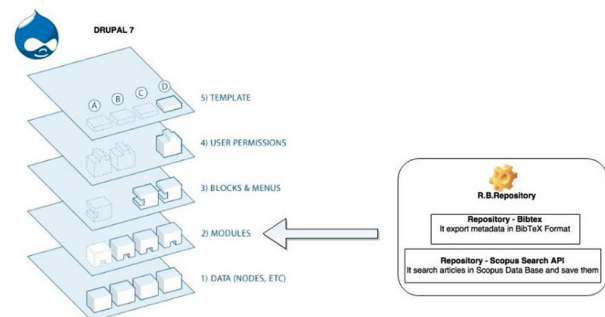


FIGURE 1. System architecture / Arquitectura del sistema

revisaron a profundidad, pero todos contenían funcionalidades más allá de lo necesario. Con la idea de no sobredimensionar el sistema y realizar un módulo independiente en la medida de lo posible, se optó por diseñar las funciones que lo caracterizan.

El módulo desarrollado está basado en la arquitectura de Drupal 7, con el propósito de ampliar las funciones que ofrece este CMS. Por ésta razón, la arquitectura del módulo está estrechamente ligada a la que presenta Drupal, como se muestra en la **FIGURA 1**.

Debido a que la finalidad es tener un repositorio de referencias bibliográficas estructurado para facilitar el manejo de la información, fue necesario definir los tipos de documentos que se manejarán y los metadatos que se asociarán a cada uno. Otro punto importante son los clientes del sistema, que, para lograr fiabilidad de la información, se decidió serían dos: administrador y usuario anónimo.

Para especificar los tipos de publicaciones se hizo una revisión de diferentes bases de datos conocidas. Finalmente se decidió hacer un grupo de siete tipos de publicaciones: artículo, libro, capítulo de libro, artículo de conferencia, tesis, patente y software, todas ellas compatibles con la clasificación reconocida por el Departamento Administrativo de Ciencia, Tecnología e Innovación de Colombia [Colciencias], como productos de la investigación.

En cuanto a los metadatos que se manejan para cada uno de estos tipos de publicación, el proceso fue similar al anterior, se buscó la información que se solicita en los diferentes estilos de referencias bibliográficas –APA, IEEE, Vancouver, entre otros–, para determinar los campos en común y fijar los datos indispensables para realizar la mayoría de estas referencias con la información que se almacena en este repositorio.

Con los resultados obtenidos se definieron los datos mínimos que debía tener cada publicación para que se almacene y procese adecuadamente en este sistema. Los resultados se muestran en las **TABLAS 1 y 2**.

IV. Design and implementation

For the design of the repository, the information related work collected during the documentation process was taken into account. Related modules were reviewed in depth, but all of them contained functionality beyond what is necessary. With the idea of not overstate the system and perform a separate module as far as possible, it was chosen to design the features that characterize it.

The developed module is based on the architecture of Drupal 7, with the purpose of expanding the features offered by this CMS. For this reason, the module architecture is closely linked to that presented in Drupal, as shown in **FIGURE 1**.

Because the purpose is to have a structured repository of bibliographic references to facilitate information management, it was necessary to define the types of documents to be managed and metadata to be associated with each one. Another important point are the users of the system, in order to achieve reliability on the information, it was decided it would be two types: administrator and anonymous user.

To specify the types of publications, a review of different known databases was made. Finally, it was decided to make a group of seven types of publications: article, book, book chapter, conference paper, thesis, patent and software, all compatible with the classification recognized by the Administrative Department of Science, Technology and Innovation of Colombia [Colciencias], as research products.

Table 1. Metadata by type of publication – mandatory fields / Metadatos por tipo de publicación – campos obligatorios

Article/Artículo	Book/Libro	Book chapter/ Capítulo de libro	Conference paper/ Ponencia en conferencia	Thesis/Tesis	Patent/Patente	Software
Title/Título	Title/Título	Book title / Título del libro Chapter title / Nombre del capítulo	Title Título	Title/Título	Invention / Invención	Name/Nombre
Authors Autor(es)	Authors Autor(es)	Authors /Autor(es)	Authors /Autor(es)	Authors Autor(es)	Inventor(s) /Inventor(es)	Producer(s) /Productor(es)
Publication year Año de publicación	Publication year Año de publicación	Publication year /Año de publicación	Publication year /Año de publicación	Publication year Año de publicación	Publication year /Año de publicación	Publication year /Año de publicación
			Conference name /Nombre de la conferencia Start of conference (year) /Inicio de la conferencia (año) End of conference (year) /Final de la conferencia (año)			
				Degree type Tipo de grado		

Table 2. Metadata by type of publication – optional fields / Metadatos para cada tipo de publicación - campos opcionales

Article/Artículo	Book/Libro	Book chapter/ Capítulo de libro	Conference paper/ Ponencia en conferencia	Thesis/Tesis	Patent/Patente	Article/Artículo
Abstract/ Resumen	Abstract/ Resumen		Abstract/ Resumen		Abstract/ Resumen	
Keywords/ Palabras clave			Keywords/ Palabras clave	Keywords/ Palabras clave		
	Sub-title/ Subtítulo					
Journal's name/ Nombre de la revista						
Journal's volumen/ Volumen de la revista	Volume/ Volumen	Volume/ Volumen				
Journal's Issue/ Edición de la revista	Issue/ Número	Issue/ Número				
First page/ Página inicial		First page/ Página inicial	First page/ Página inicial			
Last page/ Página final		Last page/ Página final	Last page/ Página final			
	Place of publication/ Lugar de publicación	Place of publication/ Lugar de publicación				Place of publication/ Lugar de publicación
	Publisher/ Editorial	Publisher/ Editorial		Academic entity/ Entidad académica		Version/ Versión
		Chapter number/ Número del capítulo	Number of conference/ Número de la conferencia			
ISSN	ISSN	ISSN				
	ISBN	ISBN				
URL	URL	URL	URL	URL	URL	URL
Doi	Doi	Doi	Doi			
Publication (day)/ Publicación (día)			Publication (day)/ Publicación (día)	Publication (day)/ Publicación (día)	Publication (day)/ Publicación (día)	
Publication (month)/ Publicación (mes)			Publication (month)/ Publicación (mes)	Publication (month)/ Publicación (mes)	Publication (month)/ Publicación (mes)	
	Editor's name/ Nombre del editor	Editor's name/ Nombre del editor				
	Language/ Idioma					
				Knowledge area/ Disciplina		
					Owner/ Propietario	
					Patent type/ Tipo de patente	
					Number/ Número	
			Conference place Lugar del evento			
			Sponsor(s) Patrocinador(es)			
			Conference (start day) Inicio de la conferencia (día)			
			Conference (start month) Mes de inicio de la conferencia			
			End of conference (day) Final de la conferencia (día)			
			End of conference (month) Final de la conferencia (mes)			

Drupal tiene inmerso entre sus funciones la gestión de usuarios: administrador es aquel que tiene control de los datos; usuario anónimo es un cliente capaz de ver la información que el administrador publica. En este sistema se manejan usuarios y publicaciones, los usuarios son los autores de dichas publicaciones.

Finalmente se obtuvo un módulo con dos submódulos: el módulo base, llamado “R. B. Repository” (Reference Bibliographies Repository) y sus dependientes, “Repository – BibTex” y “Repository – Scopus Search API”.

Las funciones principales que se disponen son cuatro:

- Gestión de información: el módulo permite agregar, editar y/o eliminar información de usuarios y publicaciones por parte del administrador.
- Navegación: se cuenta con diferentes listados para acceder a los metadatos de las publicaciones y la información de usuarios.
- Importación: para los usuarios que entre su información tengan el ID de autor asignado por Scopus, se buscará y almacenará semiautomáticamente la información de artículos que se encuentren en esa base de datos.
- Exportación: los metadatos de las publicaciones se pueden ver en formatos RIS y BibTeX.

R. B. Repository

Se encarga de crear las tablas requeridas en la base de datos, los formularios para agregar/editar cada tipo de contenido, los menús de navegación y el formato RIS para ex-

As for the metadata that are managed by types of publication, the process was similar to the above, the information requested was searched in different styles of bibliographic references –APA, IEEE, Vancouver, among others–, to determine fields in common and set data needed to perform most of these references with the information stored in this repository.

With the results obtained, the minimum data that should have each publication to be stored and properly processed in this system were defined. The results are shown in TABLES 1 and 2.

Drupal has set among its functions the user management: administrator is the one who has control of data; anonymous user is a customer able to see the information published by the administrator. In this system users and publications are managed, users are the author of such publications.

Finally, a module with two sub-modules was obtained: the base module, called “R. B. Repository” (Reference Bibliographies Repository) and its dependents, “Repository – BibTex” and “Repository – Scopus Search API”.

The main functions that are available are four:

- Information management: The module allows to add, edit or delete user information and publications by the administrator.

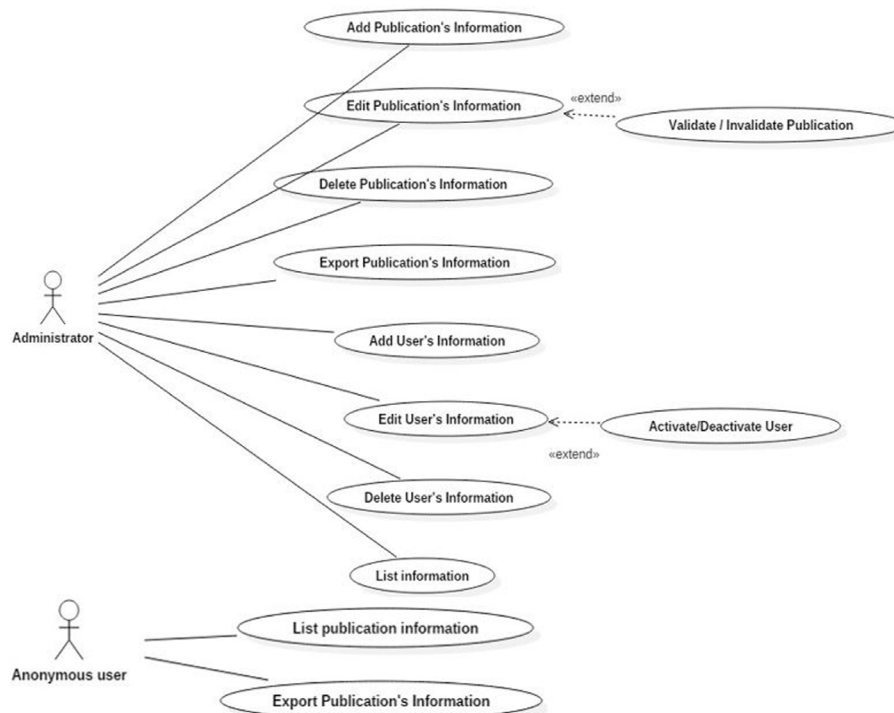


FIGURE 2. Use case diagram for R.B. Repository (base module) / Diagrama de caso de uso para el R.B.Repository (módulo base)

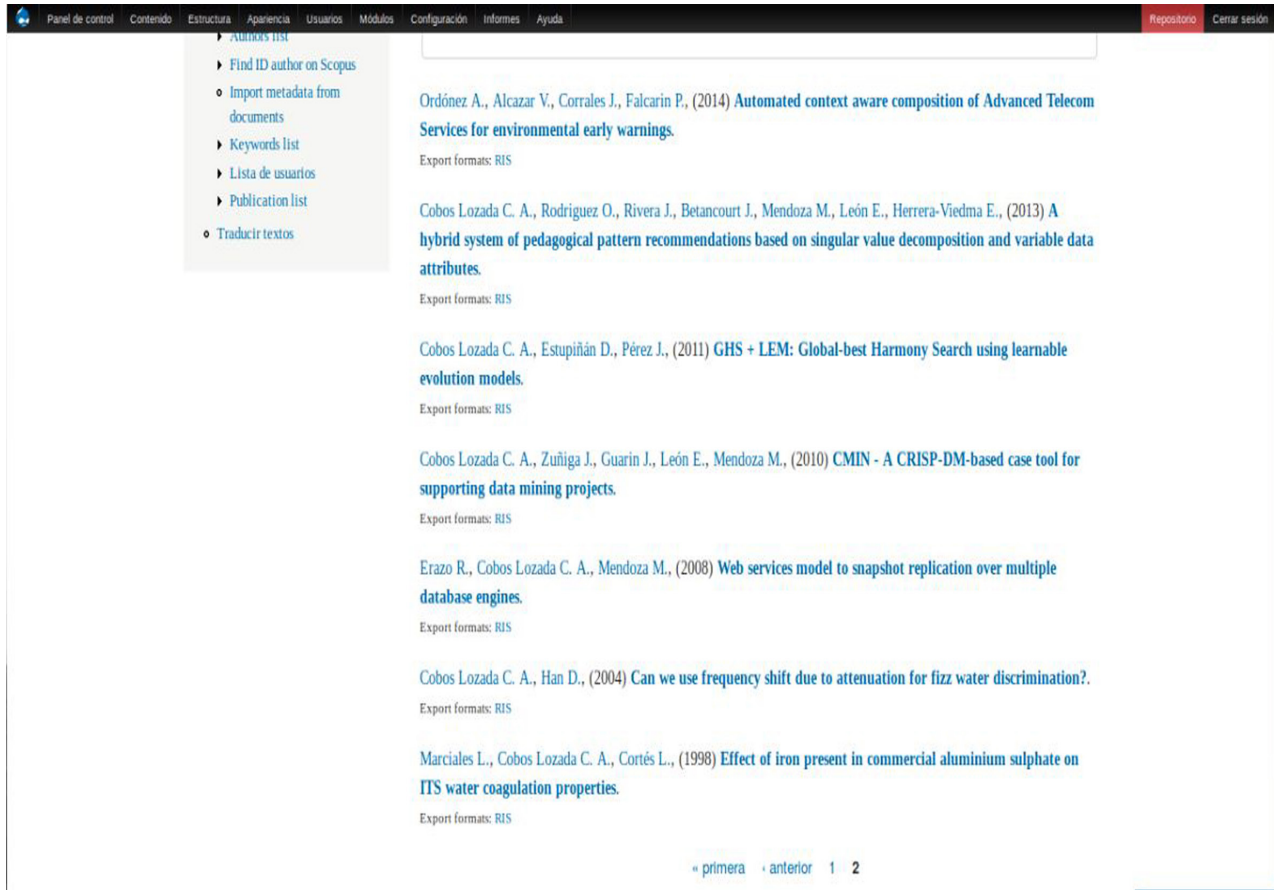


FIGURE 3. Graphical user interface for publications / Interfaz gráfica de usuario para publicaciones

- Navigation: It has different lists to access metadata publications and user information.
- Import: For users whose information have the author ID assigned by Scopus, it will be searched and semiautomatically stored the information of articles that are in that database.
- Export: Metadata publications can be viewed on RIS and BibTeX formats.

R. B. Repository

It is responsible for creating the required tables in the database, forms to add/edit each type of content, navigation menu and RIS format to export metadata of a publication. The functions that are available with this base module are shown in **FIGURE 2**.

To add a user, it is requested: full name, affiliation, email, ORCID and Scopus author ID, although not all of these data are mandatory.

The administrator has access to all the information in the database and the ability to manage it as needed, while the anonymous user can only see posts that have been validated by the administrator, as well as the list of authors and keywords.

portar los metadatos de una publicación. Las funciones de las que se dispone con este módulo base se pueden apreciar en la **FIGURA 2**.

Para agregar un usuario se solicita: nombres, apellidos, afiliación, correos electrónicos, ORCID e ID de autor en Scopus, aunque no todos estos datos son obligatorios.

El administrador tiene acceso a toda la información existente en la base de datos y la posibilidad de gestionarla según la necesidad, mientras que el usuario anónimo solo puede ver las publicaciones que han sido validadas por el administrador, además de las listas de autores y palabras clave.

La forma en que se muestra la lista de publicaciones a un usuario anónimo es: autores, año de publicación y título. En la parte inferior de esta información se encuentran los enlaces que permiten obtener todos los metadatos almacenados de la publicación en formato RIS, como se aprecia en la **FIGURA 3**.

Para el administrador se ofrecen dos listas adicionales, creadas con el fin de facilitar la identificación de las publicaciones. En estos casos no aparece la opción de exportar directamente desde la lista:

- Lista de publicaciones: expone todas las publicaciones validadas y, al final, las que aún no han sido verificadas por el administrador, con un subtítulo que permite separarlas e identificarlas.

- Lista por tipo de publicación: muestra una lista organizada según el tipo de publicación, con un encabezado que las identifica. Además de la información que se ha mencionado, también tiene el “validado” o “no validado” después del título y entre paréntesis.

En todas las listas los autores y el título de publicación son enlaces que amplían la información de la opción elegida. En el caso de los autores, se muestra el identificador del autor en el repositorio, nombres y apellidos, la cantidad de publicaciones asociadas y el título de cada una de ellas.

Para el caso particular de las publicaciones se tienen dos opciones para el despliegue de información. En la lista de publicaciones a la que acceden los usuarios anónimos, se muestran los datos según lo dicho en las **TABLAS 1 y 2**, y la opción de exportarlos en formato RIS; si se accede a través de las listas dadas al administrador, además de lo anterior, se presentan tres botones: editar, eliminar y validar o quitar la validación.

En ambos casos los autores y las palabras clave son enlaces que extienden la información ofrecida según la selección. Una palabra clave muestra el número de publicaciones relacionadas con ella y el título de cada una; caso similar a la descripción dada si se elige un autor.

Para listar los usuarios se dispone de tres opciones con el fin de facilitar el trabajo del administrador, ya que esta información no está disponible para usuarios anónimos. El estado activo representa usuarios que están vinculados a la entidad que está empleando el repositorio; el estado inactivo es para aquellos que estuvieron relacionados, pero que en ese momento no son parte de la organización, y no se requiere seguir ampliando su información. Este es un parámetro importante en el módulo Repository – Scopus Search API.

- Lista de usuarios: listado general de todos los usuarios que existen en la base de datos del repositorio; se presenta en una tabla que contiene ID de usuario, primer nombre, apellidos, correo electrónico principal y estado.
- Lista de usuarios activos: como su nombre lo indica, es un listado de usuarios en estado activo; esta tabla muestra ID, primer nombre, apellidos y correo electrónico.
- Lista de usuarios inactivos: tiene la misma disposición que la lista de usuarios activos, con la salvedad de que en este caso los usuarios tendrán asociado el estado inactivo.

Las dos últimas listas descritas ofrecen además la opción de cambiar el estado de uno o más usuarios de activo a inactivo y de inactivo a activo, respectivamente. La lista de usuarios inactivos también da la opción de eliminar usuarios definitivamente.

Todas estas listas tienen un hipervínculo en el ID que dirige a la información detallada del usuario, donde se tiene la opción de editar, desactivar o eliminar.

The way the list of publications is shown to an anonymous user is as follows: authors, year of publication and title. At the bottom of this information there are links that allow to obtain all metadata stored of the publication in RIS format, as shown in **FIGURE 3**.

For the manager it is offered two additional lists, created in order to facilitate the identification of publications. In these cases, the option to export directly from the list does not appear:

- List of publications: It exposes all validated publications and, in the end, those that have not yet been verified by the administrator, with a caption that allows to separate and identify them.
- List by publication type: Displays an organized list by type of publication, with a header that identifies them. In addition to the information mentioned, it also has “validated” or “not validated” after the title and in parentheses.

In all lists, the authors and the title of publication are links that provide further information on the option chosen. In the case of the authors, it is shown the author identifier in the repository, full names, number of associated publications and title of each one.

For the particular case of publications, there are two options for the display of information. In the list of publications where the anonymous users have access, it is shown the data as mentioned in Tables 1 and 2, and the option to export them in RIS format; if it is accessed through the lists given to the administrator; in addition to the above, three buttons appear: edit, delete and validate or remove validation.

In both cases the authors and keywords are links that provide further information depending on the selection. A keyword shows the number of publications related to it and the title of each one; similar case to the description given if an author is chosen.

To list the users, there are three options in order to facilitate the work of the administrator because this information is not available for anonymous users. The active state represents users who are linked to the entity that is using the repository; the inactive state is for those who were related but in that moment are not part of the organization, and providing further information is not required. This is an important parameter in the Repository module – Scopus Search API.

- List of users: General list of all users that exist in the database repository; it is presented in a table that contains user ID, first name, last name, primary email and state.

- List of active users: As its name implies, is a list of users in active state; this table shows ID, first name, last name and email.
- List of inactive users: It has the same disposition as the list of active users, except that in this case users will have associate the inactive state.

The last two lists described, also offer the option to change the status of one or more users from active to inactive and from inactive to active, respectively. The list of inactive users also gives the option to delete users permanently.

All these lists have a hyperlink in the ID that redirects to the detailed user information, where there is an option to edit, deactivate or delete.

Repository – BibTex

This module offers the option to export the metadata of publications in BibTeX format. It is responsible for transforming plain text information to this new format, in which special characters, such as a tilde, for example, are represented under specific parameters. The option is available on the list and in the detailed information of publications, and is located next to RIS format.

Repository – Scopus Search API

With this module two functions are obtained, one responsible for finding the author ID in Scopus for users who do not have it in the database and another is responsible for searching, processing and storing or editing information of publications with the author ID of users who have registered it, all through Scopus API.

The search for authors was developed in order to detect matches between the user name registered in the repository and authors in the Scopus database. In case of finding them, it is deployed a tabular list where the author ID, full name and affiliation data registered in the system are available.

Each of these tables is headed by the full name of the user registered in the repository, to avoid confusion. If the administrator finds that one of the listed names corresponds to user interest, the process of association of author ID given by Scopus and the repository user can be started.

This first feature assists the second, since without an author ID for Scopus, document search is not performed on that or those users. It works under this requirement to avoid introducing unwanted information to the system, which may be given by homonyms.

Repository – BibTex

Este módulo ofrece la opción de exportar los metadatos de las publicaciones en formato BibTeX. Se encarga de transformar la información de texto plano a este nuevo formato, en el cual los caracteres especiales, como una tilde, por ejemplo, son representados bajo parámetros específicos. La opción está disponible en la lista y en la información detallada de las publicaciones, y se ubica al lado del formato RIS.

Repository – Scopus Search API

Con este módulo de obtienen dos funciones, una encargada de buscar el ID de autor en Scopus para los usuarios que no lo tengan en la base de datos y otra encargada de buscar, procesar y almacenar o editar información de publicaciones con el ID de autor de los usuarios que lo tengan registrado, todo ello a través de la API de Scopus.

La búsqueda de autores se desarrolló con el fin de detectar coincidencias entre el nombre del usuario registrado en el repositorio y autores en la base de datos de Scopus. En caso de encontrarlas, se despliega una lista tabulada donde se dispone de la ID de autor, su nombre completo y los datos de afiliación registrados en su sistema.

Cada una de estas tablas está encabezada por el nombre completo del usuario registrado en el repositorio, para evitar confusiones. Si el administrador encuentra que uno de los nombres listados corresponde al usuario de interés, puede iniciar el proceso de asociación del ID de autor dado por Scopus y el usuario del repositorio.

Esta primera función asiste a la segunda, ya que si no se tiene un ID de autor para Scopus, la búsqueda de documentos no se realizará sobre ese o esos usuarios. Funciona bajo este requisito para evitar introducir información no deseada al sistema, que puede estar dada por homónimos.

Con el ID de autor, y por medio de la API de Scopus, se puede extraer información de las publicaciones verificadas por esta reconocida base de datos. Para esta primera versión del módulo se procesará información de artículos, debido a que está entre los tipos de publicación más comunes en el entorno que da cabida a este proyecto.

El funcionamiento está representado en el diagrama de la **FIGURA 4**. Para cada usuario se realiza una búsqueda de las publicaciones asociadas a él en la base de datos de Scopus a

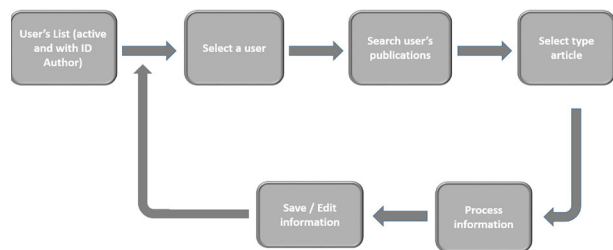


Figure 4. Metadata search process with Scopus API / Proceso de búsqueda de metadatos con la API de Scopus

través de su API, de esta lista se seleccionan solo los artículos para almacenarlos en el repositorio o editarlos en caso de encontrar nueva información que complemente la existente.

Esta API funciona a través de las URL, por lo que es necesario realizar más de una solicitud para obtener la mayor cantidad de información posible en lo que concierne a una publicación. El bloque *Process information* de la **FIGURA 4**, representa encontrar una publicación del tipo artículo y realizar las peticiones adicionales para obtener la información requerida. Para los artículos se busca información para todos sus campos, exceptuando palabras clave y URL.

V. Análisis de resultados

Con el módulo desarrollado, una persona o entidad que necesite que su página web mantenga actualizados los metadatos de publicaciones ahorraría mucho tiempo en el proceso de gestión de información; además, si cuenta con acceso a Scopus la velocidad de adquisición y almacenamiento de las publicaciones tipo artículo se incrementaría sustancialmente.

El ingreso manual de la información no alcanza la velocidad del servicio semiautomático que se ofrece, pero facilita el trabajo de la persona o personas encargadas de esta labor. Si los datos son ingresados manualmente la publicación será validada de forma automática, pero si se adquirió a través de la API el administrador debe validarla para que sea visible por un usuario anónimo.

El proceso manual de información se comprobó empleando dos sujetos de prueba que ingresaron diferentes tipos de publicación y posteriormente las editaron y exportaron. Este proceso se realizó en tres ocasiones para corregir secciones de los formularios que ellos encontraron confusas.

Para analizar el funcionamiento del módulo “Repository – Scopus Search API” se realizaron cinco pruebas independientes. Cada petición a través de la API trae todas las publicaciones que tienen relación con el autor sobre el que se realiza la búsqueda, el sistema procesa todos estos documentos para identificar solo los artículos y finalmente procesa esta información para almacenarla adecuadamente. Por esta razón se muestran los resultados en función de la cantidad de artículos recuperados.

En las **TABLAS 3 a 7** se presentan diferentes casos que tienen relación con la presencia y ausencia de información en los diferentes campos. La A representa a la API y la S a la página de Scopus, si tienen una comilla (?), quiere decir que se requiere aclaración, por ejemplo, que no para todos los artículos se encontró ese dato. En caso de aparecer en rojo, significa que el dato no fue encontrado o no es proporcionado por la herramienta.

En la **TABLA 3** se procesaron diecisiete publicaciones, entre las que se tenían siete clasificadas como artículos; en la **TABLA 4** veintidós publicaciones y trece artículos; en la **TABLA 5** treinta y tres publicaciones y diez artículos; en la **TABLA 6** treinta y cuatro publicaciones y diecisiete artículos; y en la **TABLA 7** cincuenta publicaciones y once artículos.

With author ID, and through Scopus API, information from verified publications by this renowned database can be extracted. For this first version of the module, information of articles will be processed because it is among the most common types of publication in the environment that accommodates this project.

The operation is represented in the diagram in **FIGURE 4**. For each user a search for publications associated with him or her is performed in the Scopus database through its API, from this list are selected only articles to be stored in the repository or edit them in case of finding new information that complements the existing one.

This API works through URLs, so it is necessary to make more than one application to obtain as much information as possible what respect to a publication. Process information block in **FIGURE 4**, depicts finding a publication of the article type and making additional requests to obtain the required information. For articles, information for all fields is searched excepting keywords and URL.

V. Analysis of results

With the developed module, a person or entity that needs that their website keeps updated publications metadata would save a lot of time in the process of information management; also if there is open access to Scopus, acquisition speed and publications storage of article type would increase substantially.

The manual entry of information does not reach the speed of semiautomatic service offered, but facilitates the work of the person or persons responsible for this work. If data are entered manually so the publication will be validated automatically, but if it was acquired through the API the administrator must validate it to make it visible for an anonymous user.

The manual process of information was checked using two test subjects that entered different types of publication and subsequently were published and exported. This process was performed three times to correct sections of the forms they found confusing.

To analyze the operation of the “Repository – Scopus Search API” module, five independent tests were performed. Each request through the API brings all publications that are related to the author on which the search is performed, the system processes all these documents to identify only the articles and finally processes this information to store it properly. For this reason the results are displayed based on the number of retrieved articles.

Table 3. Seven articles in seventeen documents / Siete artículos entre diecisiete documentos

		Caso 1	Caso 2	Caso 3	Caso 4
Type / Tipo de documento		A S	A S	A S	A S
Title / Título		A S	A S	A S	A S
Authors / Autores		A S	A S	A S	A S
Abstract / Resumen		A S	A S	A S	A S
Keywords / Palabras clave		A' S	A' S	A' S	A' S
Publication date / Fecha de publicación	Day/ Día	A S	A S	A S	A S
	Month / Mes	A S	A S'	A S	A S
	Year / Año	A S	A S	A S	A S
Journal / Revista		A S	A S	A S	A S
Volume / Volumen		A S	A S	A S	A S
Issue / Edición		A S	A S	A S	A S
First page / Página inicial		A S	A S	A S	A S
Last page / Página final		A S	A S	A S	A S
ISSN		A S	A S	A S	A S
URL		A' S	A' S	A' S	A' S
Doi		A S	A S	A S	A S
Total		3	2	1	1

Table 5. Ten articles in thirty three documents / Diez artículos entre treinta y tres documentos

		Caso 1	Caso 2	Caso 3	Caso 4
Type / Tipo de documento		A S	A S	A S	A S
Title / Título		A S	A S	A S	A S
Authors / Autores		A S	A S	A S	A S
Abstract / Resumen		A S	A S	A S	A S
Keywords / Palabras clave		A' S	A' S	A' S	A' S
Publication date / Fecha de publicación	Day/ Día	A' S'	A S	A S	A' S'
	Month / Mes	A' S'	A S'	A S	A S
	Year / Año	A S	A S	A S	A S
Journal / Revista		A S	A S	A S	A S
Volume / Volumen		A S	A S	A S	A S
Issue / Edición		A S	A S	A S	A S
First page / Página inicial		A S	A S	A S	A S
Last page / Página final		A S	A S	A S	A S
ISSN		A S	A S	A S	A S
URL		A' S	A' S	A' S	A' S
Doi		A S	A S	A S	A S
Total		6	2	1	1

In TABLES 3 to 7, different cases that are related to the presence and absence of information in the different fields are presented. A represents API and S represents Scopus page, if they have a quotation mark ('), it means that clarification is required, for example, not for all articles that datum was found. In case of appearing in red, it means that datum was not found or is not provided by the tool.

In TABLE 3, were processed seventeen publications among which seven were classified as articles; in TABLE 4, twenty two publications and thirteen articles; in TABLE 5, thirty-three publications and ten articles; in TABLE 6, thirty-four publications and seventeen articles; and in TABLE 7, fifty publications and eleven articles.

In the first case of TABLE 3, it was found the particularity that the three articles found had been published in the

Table 4. Thirteen articles in twenty two documents / Trece artículos entre veintidós documentos

		Caso 1	Caso 2	Caso 3	Caso 4
Type / Tipo de documento		A S	A S	A S	A S
Title / Título		A S	A S	A S	A S
Authors / Autores		A S	A S	A S	A S
Abstract / Resumen		A S	A S	A S	A S
Keywords / Palabras clave		A' S	A' S	A' S	A' S
Publication date / Fecha de publicación	Day/ Día	A S	A S	A S	A S
	Month / Mes	A S	A S'	A S	A S
	Year / Año	A S	A S	A S	A S
Journal / Revista		A S	A S	A S	A S
Volume / Volumen		A S	A S	A S	A S
Issue / Edición		A S	A S	A S	A S
First page / Página inicial		A S	A S	A S	A S
Last page / Página final		A S	A S	A S	A S
ISSN		A S	A S	A S	A S
URL		A' S	A' S	A' S	A' S
Doi		A S	A S	A S	A S
Total		8	2	2	1

Table 6. Seventeen articles in thirty four documents / Diecisiete artículos entre treinta y cuatro documentos

		Caso 1	Caso 2	Caso 3	Caso 4	Caso 5
Type / Tipo de documento		A S	A S	A S	A S	A S
Title / Título		A S	A S	A S'	A S	A S
Authors / Autores		A S	A S	A S	A S	A S
Abstract / Resumen		A S	A S	A S	A S	A S
Keywords / Palabras clave		A' S	A' S	A' S	A' S	A' S
Publication date / Fecha de publicación	Day/ Día	A S'	A S'	A S'	A S'	A S
	Month / Mes	A S'	A S'	A S	A S	A S
	Year / Año	A S	A S	A S	A S	A S
Journal / Revista		A S	A S	A S	A S	A S
Volume / Volumen		A S	A S	A S	A S	A S
Issue / Edición		A S	A S	A S	A S	A S
First page / Página inicial		A S	A S	A S	A S	A S
Last page / Página final		A S	A S	A S	A S	A S
ISSN		A S	A S	A S	A S	A S
URL		A' S	A' S	A' S	A' S	A' S
Doi		A S	A S	A S	A S	A S
Total		7	3	3	2	2

Table 7. Eleven articles in fifty documents / Once artículos entre cincuenta documentos

		Caso 1	Caso 2	Caso 3	Caso 4
Type / Tipo de documento		A S	A S	A S	A S
Title / Título		A S	A S	A S	A S
Authors / Autores		A S	A S	A S	A S
Abstract / Resumen		A S	A S	A S	A S
Keywords / Palabras clave		A' S	A' S	A' S	A' S
Publication date / Fecha de publicación	Day/ Día	A S'	A S	A S	A S
	Month / Mes	A S'	A S	A S	A S
	Year / Año	A S	A S	A S	A S
Journal / Revista		A S	A S	A S	A S
Volume / Volumen		A S	A S	A S	A S
Issue / Edición		A S	A S	A S	A S
First page / Página inicial		A S	A S	A S	A S
Last page / Página final		A S	A S	A S	A S
ISSN		A S	A S	A S	A S
URL		A' S	A' S	A' S	A' S
Doi		A S	A S	A S	A S
Total		8	1	1	1

En el caso 1 de la **TABLA 3** se encontró la particularidad de que los tres artículos encontrados habían sido publicados en la misma revista y ésta solo maneja volumen no número, por lo que esta información no se obtuvo ni a través de la API ni desde Scopus. El campo URL se había definido que se utilizaría para la ubicación del texto completo preferiblemente, la API ofrece una URL asociada a Scopus y la intención no es redirigir al usuario a otra página con metadatos. Esto último sucedió en todos los casos.

En la **TABLA 6** se detectó una publicación duplicada en Scopus. El mismo título, clasificado en un caso como “Article” y en otro como “Conference Paper”. Después de verificar la información suministrada, que era casi idéntica en los dos casos, se determinó que correspondía al tipo “Conference Paper” y por tanto, sería un archivo indeseado en el tipo “Artículo” del repositorio.

Todos los artículos fueron recuperados y no se detectaron errores en el proceso de extracción de la información, pero, como se evidencia en las tablas, no toda la información fue veraz, principalmente en lo que a las fechas de publicación se refiere. De hecho, las fechas de publicación fueron el mayor inconveniente detectado, ya que para todos los artículos recuperados con la API se tenía día, mes y año, pero desde la página de Scopus, en la mayoría de los casos, solo se mostraba el año y en muy pocas ocasiones la fecha completa. Durante el proceso de verificación se encontraron varios datos de mes que no correspondían a la fecha de publicación de la revista o libro y casi todos los datos del día no pudieron ser verificados. Esta podría ser una de las razones por las que la fecha en la página de Scopus, por lo general, se limita al año.

Es posible acceder a información de las palabras clave desde la API, pero todas las pruebas que se realizaron obtenían solo una palabra clave, sin embargo, desde la página de Scopus se puede ver, no solo las palabras clave del artículo, sino también las asociadas al autor. Por ésta razón se decidió no implementar ésta parte en el módulo, ya que es un llamado independiente y se estaría incrementando la cantidad de procesamiento necesario sin buenos resultados.

En cuanto a la función de exportar, tanto para el formato RIS, como para el BibTeX, los resultados fueron los esperados. La información que obtenía el usuario era la que se encontraba hasta el momento en la base de datos, sin considerar campos que no tuvieran información.

VI. Conclusiones y trabajo futuro

Con los resultados obtenidos se soluciona una parte del problema que se presenta en el portal del Departamento de Telemática de la Universidad del Cauca. En caso de utilizar el módulo R. B. Repository, la adquisición de los metadatos de publicaciones tendría un componente semiautomático lo que reduciría la cantidad de información que se debería introducir manualmente.

same journal and it only manages volume not number, so this information was not obtained either through API or Scopus. The URL field had been defined that would be used for the location of the full text preferably, the API provides a URL associated to Scopus and it is not intended to redirect the user to another page with metadata. The latter occurred in all cases.

In **TABLE 6**, a duplicate publication in Scopus was detected. The same title, in a case classified as “Article” and another as “Conference Paper”. After verifying the information provided, which was almost identical in both cases, it was determined that corresponded to “Conference Paper” type and therefore it would be an unwanted file in “Article” type of the repository.

All articles were recovered and no errors were detected in the process of extracting information, but, as evidenced in the tables, not all information was accurate, especially as far as release dates are concerned. In fact, the publication dates were the biggest drawback detected, since for all articles recovered with the API data such as day, month and year were available, but from the Scopus page, in most cases, only year and rarely full date was shown. During the verification process it was found that several month data did not correspond to the date of publication of the journal or book and almost every day data could not be verified. This could be one of the reasons why the date in Scopus page is usually limited to year.

It is possible to access to keywords information from the API, but all tests performed got only a keyword, however, from the Scopus page is possible to see not only the keywords in the article, but also the ones associated to the author. For this reason it was decided not to implement this part on the module, considering that it is an independent appeal and would increase the amount of processing required without good results.

As for the export function, for both the RIS format and for BibTeX, the results were as expected. The information obtained by the user was the one that was currently in the database, without considering fields that not had information.

VI. Conclusions and future work

With the results obtained, a part of the problem presented on the website of Department of Telematics of the University of Cauca is solved. When using the R. B. Repository module, acquiring metadata publications will

have a semiautomatic component which would reduce the amount of information that should be entered manually.

The module developed did not present problems in the installation and uninstallation processes defined by Drupal 7. In the first process all desired functions and parameters were established, and in the second process were eliminated without interfering with the system.

As for the manual operation of the repository, processes requiring additional steps have annotations to guide the administrator; semiautomatic part managed to get all articles among all possible publications retrieved through the API.

The most recurrent problem was found in the publication dates, because the only consistent parameter was the year, when it comes to veracity of information. Instead, the month and day retrieved did not always correspond to the publication of the book or journal that contained them.

The module is able to find all types of publication associated with an author. However, a version that achieves to store information publication types such as book, book chapter and conference paper could be performed, that are available in Scopus, and also searches could be added in other databases such as Google Scholar that has lots of information.

Scopus API offers many possibilities of acquiring information, so that could be considered for a new version, another way to obtain the date of publication, to confirm that the data are reliable or correspond to what is visible from Scopus page, if it is presented is because it has been validated. *sr*

El módulo desarrollado no presentó inconvenientes en los procesos de instalación y desinstalación definidos por Drupal 7. En el primer proceso se crearon todas las funciones y parámetros deseados, y en el segundo se eliminaron sin interferir con el sistema.

En cuanto al funcionamiento manual del repositorio, los procesos que requerían de pasos adicionales presentan anotaciones para guiar al administrador; la parte semiautomática logró obtener la totalidad de artículos entre todas las publicaciones posibles que se recuperaban a través de la API.

El problema más recurrente se encontró en las fechas de publicación, debido a que el único parámetro consistente en cuanto a veracidad de información fue el año. En cambio, el mes y el día recuperados no siempre correspondían al de la publicación del libro o revista que los contenía.

El módulo es capaz de encontrar todo tipo de publicación asociada a un autor. Sin embargo, podría realizarse una versión que logre almacenar información de tipos de publicación como libro, capítulo de libro y artículo de conferencia, que se encuentran en Scopus, y también se podría agregar búsquedas en otras bases de datos como Google Scholar que posee gran cantidad de información.

La API de Scopus ofrece muchas posibilidades de adquisición de información, por lo que se podría considerar, para una nueva versión, otra forma de obtener la fecha de publicación, para afirmar que los datos obtenidos son confiables y/o corresponden a lo que es visible desde la página de Scopus, que si está presente es porque se ha validado. *sr*

References / Referencias

- Bibliography Module*. (2006, June 17). Retrieved from: <https://www.drupal.org/project/biblio>
- Book Post*. (2008, October 7). Retrieved from: <https://www.drupal.org/project/bookpost>
- Chacón, M., & Cruz, J. (2012). Metabuscador académico para referencias bibliográficas. In: *Proceedings of the SPDECE-2012. Ninth multidisciplinary symposium on the design and evaluation of digital content for education* (pp. 221-232). Alicante, España: Universidad de Alicante.
- Chua, C., Cao, L., Cousins, K., Mohan, K., Straub, D., & Vaishnavi, V. (2002). IS bibliographic repository (ISBIB): A central repository of research information for the IS community. *Communications of the Association for Information Systems*, 8(1), 392-412.
- [D7] Onki*. (2012, May 28). Retrieved from: <https://www.drupal.org/node/1604784>
- Dicheva, D., & Dichev, C. (2014). Finding open educational resources in computing. In: *Advanced Learning Technologies (ICALT)*, 2014 IEEE 14th International Conference on. IEEE.
- Easy import*. (2011, September 4). Retrieved from: <https://www.drupal.org/node/1269194>
- Feitelson, D. (2000). Cooperative indexing, classification, and evaluation in BoW. In: *Cooperative Information Systems* (pp.66-77). Berlin-Heidelberg: Springer.
- Flores, G., & Sánchez, N. (2007). Los repositorios institucionales: análisis de la situación internacional y principios generales para Cuba. *ACIMED*, 16(6). Retrieved from http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352007001200006
- Import/Export*. (2009, March 17). Retrieved from: https://www.drupal.org/project/import_export_tool
- Iskme*. (2007). *OER Commons*. Retrieved from: <https://www.oercommons.org/>
- Masterkey: Drupal (MkDru)*. (2014, January22). Retrieved from: <http://www.indexdata.com/masterkey-drupal>
- OAI-PMH Module*. (2006, August 5). Retrieved from: <https://www.drupal.org/project/oi2>
- Serrano, C. (2005). *Modelo integral para el profesional en ingeniería*. Popayán, Colombia: Universidad del Cauca.
- Simple CSV Importer*. (2015, March 11). Retrieved from: <https://www.drupal.org/sandbox/redorbit/2450325>
- Taxonomy manager*. (2007, June 18). Retrieved from: https://www.drupal.org/project/taxonomy_manager
- The eXtensible Catalog (XC) Drupal Toolkit*. (2009, March 9). Retrieved from: <https://www.drupal.org/project/xc>
- Z39.50/SRU Client*. (2006, April 19). Retrieved from: <https://www.drupal.org/node/59457>

CURRICULUM VITAE

Fabián Ortiz Collazos Student of Electronics and Telecommunications Engineering (emphasis in Telecommunications Engineering) at the Universidad del Cauca (Popayán, Colombia) / Estudiante de Ingeniería en Electrónica y Telecomunicaciones (énfasis en Telecomunicaciones) de la Universidad del Cauca.

Beatriz Elena Hurtado Hurtado Student of Electronics and Telecommunications Engineering (emphasis in Telematics Engineering) at the Universidad del Cauca (Popayán, Colombia) / Estudiante de Ingeniería en Electrónica y Telecomunicaciones (énfasis en Ingeniería Telemática) de la Universidad del Cauca.

Eduardo Rojas Pineda Electronics Engineer (Universidad del Cauca, Popayán-Colombia), Master in Systems and Communications Networks (Universidad Politécnica de Madrid, España) and Master in Higher Education Management (Universidad de los Andes, Bogotá-Colombia). Full professor at the Department of Telematics, Faculty of Electronics and Telecommunications Engineering, Universidad del Cauca / Ingeniero Electrónico de la Universidad del Cauca, con Máster en Sistemas y Redes de Comunicaciones de la Universidad Politécnica de Madrid, y Máster en Dirección Universitaria de la Universidad de los Andes. Es profesor titular de tiempo completo del Departamento de Telemática de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca.