



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Diseño e implementación de un proceso de migración para  
la biblioteca Digital de SISTEDES

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Palacios Gallego, Miguel

Tutor/a: Canós Cerdá, José Hilario

Director/a Experimental: GOMEZ LLANA, ABEL

CURSO ACADÉMICO: 2022/2023

# Resumen

---

La Biblioteca Digital de SISTEDES (*BDSISTEDES*) es el repositorio institucional de la Sociedad de Ingeniería de Software y Tecnologías de Desarrollo de Software, que incluye las actas de los diversos congresos que dicha sociedad organiza anualmente, junto con otros documentos de interés. La versión actual de la biblioteca está desarrollada sobre una plataforma de gestión de contenido de propósito general, lo que supone ciertas disfunciones en la gestión general de los fondos de *BDSISTEDES*. Por ello, se plantea en este trabajo un cambio tecnológico que permita superar las disfunciones actuales. En concreto, se ha optado por realizar una migración del sistema actual al sistema Open Journal Systems (*OJS*), un gestor de contenido académico de reconocido prestigio, que cuenta con numerosas implantaciones exitosas en todo el mundo. Para ello, el trabajo consistirá en el diseño y desarrollo del proceso de migración, que incluirá la adaptación del contenido a la nueva tecnología, así como el desarrollo de todas las utilidades necesarias para la correcta transición.

**Palabras clave:** migración, biblioteca digital, *WordPress*, *OJS*, metadatos.

# Resum

---

La Biblioteca Digital de SISTEDES (*BDSISTEDES*) és el repositori institucional de la Societat d'Enginyeria de Software i Tecnologies de Desenvolupament de Software, que inclou les actes dels diversos congressos que aquesta societat organitza anualment, juntament amb altres documents d'interés. La versió actual de la biblioteca està desenvolupada sobre una plataforma de gestió de contingut de propòsit general, la qual cosa suposa unes certes disfuncions en la gestió general dels fons de *BDSISTEDES*. Per això, es planteja en aquest treball un canvi tecnològic que permeti superar les disfuncions actuals. En concret, s'ha optat per realitzar una migració del sistema actual al sistema Open Journal Systems (*OJS*), un gestor de contingut acadèmic de reconegut prestigi, que compta amb nombroses implantacions reeixides a tot el món. Per a això, el treball consistirà en el disseny i desenvolupament del procés de migració, que inclourà l'adaptació del contingut a la nova tecnologia, així com el desenvolupament de totes les utilitats necessàries per a la correcta transició.

**Paraules clau:** migració, biblioteca digital, *WordPress*, *OJS*, metadades.

# Abstract

---

The SISTEDES Digital Library (*BDSISTEDES*) is the institutional repository of the Software Engineering and Software Development Technologies Society, which includes the proceedings of the various conferences organized annually by the society, along with other interesting documents. The current version of the library is developed on a general purpose content management platform, which implies certain dysfunctions in the general management of the *BDSISTEDES* collection. Therefore, this work proposes a technological change to overcome the current dysfunctions. Specifically, it has been decided to migrate from the current system to the Open Journal Systems (*OJS*), an academic content manager of recognized prestige, which has numerous successful implementations worldwide. To this end, the work will consist of the design and development of the migration process, including the adaptation of the content to the new technology, as well as the development of all the necessary utilities for the correct transition.

**Keywords :** migration, digital library, *WordPress*, *OJS*, metadata.



# Índice general

---

CAPÍTULO 1 - INTRODUCCIÓN.....	9
1.1 Contexto.....	11
1.2 Problema.....	12
1.3 Solución.....	12
1.4 Estructura.....	13
CAPÍTULO 2 - CONTEXTO TECNOLÓGICO .....	15
2.1 <i>WordPress</i> .....	18
2.2 <i>EasyChair</i> .....	20
2.3 <i>Open Journal System (OJS)</i> .....	20
2.3.1 Ejemplos de sitios con OJS.....	21
2.4 Propuesta.....	24
CAPÍTULO 3 - ARQUITECTURA DE LA INFORMACIÓN .....	25
3.1 Origen 1: <i>BDSISTEDES</i> .....	27
3.2 Origen 2: <i>EasyChair</i> .....	29
3.3 Destino: <i>BDSISTEDES</i> en <i>OJS</i> .....	30
CAPÍTULO 4 - TECNOLOGÍA UTILIZADA.....	33
4.1 <i>Altova XMLSpy</i> .....	35
4.2 <i>Altova MapForce</i> .....	37
4.3 <i>Visual Studio Code</i> .....	38
CAPÍTULO 5 - MIGRACIÓN 1: CONTENIDO LEGADO .....	39
5.1 Descripción de la fuente de datos.....	41
5.2 Definición de la transformación .....	42
5.2.1 <i>Handles</i> .....	42
5.2.2 <i>Secciones</i> .....	44
5.2.3 <i>Archivos PDF</i> .....	46
5.2.4 <i>Autores</i> .....	47
5.3 Carga de datos.....	49



CAPÍTULO 6 - MIGRACIÓN 2: CARGA ANUAL .....	53
6.1 Descripción de la fuente de datos.....	55
6.2 Definición de la transformación .....	56
6.3 Carga de datos.....	59
CAPÍTULO 7 - CONCLUSIONES .....	61
REFERENCIAS .....	65
ANEXO I: MAPPING CARGA ANUAL.....	67
ANEXO II: OBJETIVOS DE DESARROLLO SOSTENIBLE.....	71

# Índice de figuras

---

Figura 1: Esquema de migración en BDSISTEDES.....	11
Figura 2: Logo oficial de WordPress.....	18
Figura 3: Interfaz WordPress.....	19
Figura 4: Logo oficial de EasyChair.....	20
Figura 5: Logo oficial de OJS.....	20
Figura 6: Revista UMA.....	21
Figura 7: Artículo en revista UMA.....	22
Figura 8: Portal de revistas de la UPV.....	22
Figura 9: Portal de revistas de la UV.....	23
Figura 10: Número de una revista UV.....	24
Figura 11: Interfaz en BDSISTEDES.....	28
Figura 12: Tracks en BDSISTEDES.....	28
Figura 13: Artículo en BDSISTEDES.....	29
Figura 14: Exportación de EasyChair.....	29
Figura 15: Fichero submissions.csv EasyChair.....	30
Figura 16: Estructura de BDSISTEDES en OJS.....	31
Figura 17: Estructura de las actas de conferencias en OJS.....	31
Figura 18: XML destino OJS.....	32
Figura 19: Editor XMLSpy de Altova.....	35
Figura 20: Corrección de errores con XMLSpy.....	36
Figura 21: Diseño de transformación en MapForce.....	37
Figura 22: Metadatos de BDSISTEDES.....	41
Figura 23: Metadatos de los boletines de BDSISTEDES.....	42
Figura 24: Estructura de un handle.....	43
Figura 25: Interfaz de WordPress.....	44
Figura 26: Vista interfaz de JISBD en OJS.....	45
Figura 27: Secciones en XML.....	46
Figura 28: Metadatos WordPress.....	48
Figura 29: Módulo XML Nativo.....	49
Figura 30: Vista interfaz de PROLE en OJS.....	51
Figura 31: Vista interfaz de un artículo en OJS.....	51
Figura 32: vista MapForce carga anual.....	57
Figura 33: Fichero final carga anual.....	58
Figura 34: metadatos de los papers (Anexo I).....	67
Figura 35: metadatos de los autores (Anexo I).....	68
Figura 36: keywords de las submissions (Anexo I).....	68
Figura 37: esquema completo JCIS (Anexo I).....	69

# Índice de tablas

---

Tabla 1: transformaciones de la migración de BDSISTEDES.....	49
Tabla 2: transformaciones de la migración de EasyChair .....	59
Tabla 3: Objetivos de Desarrollo Sostenible .....	71

# CAPÍTULO 1 - Introducción

---



## 1.1 Contexto

La Sociedad de Ingeniería del Software y Tecnologías de desarrollo de Software (SISTEDES) es una asociación sin ánimo de lucro, dedicada a promover la investigación, la transferencia y la innovación en esos campos. Además, contribuye al desarrollo científico y tecnológico en el área de las Tecnologías de Desarrollo de Software y de la Ingeniería de Software, ofreciendo actividades y servicios a sus afiliados y a la comunidad española que trabaja en estas áreas. También organiza jornadas científicas anuales (Jornadas SISTEDES), mantiene una biblioteca digital en la que se almacenan las actas de las Jornadas, así como otros contenidos digitales de interés para la comunidad. [SISTEDES, 2022]

Este proyecto consiste en el diseño e implementación de un proceso de migración de la biblioteca digital de SISTEDES desde su plataforma actual (*WordPress*<sup>1</sup>) a otro software, *Open Journal Systems (OJS)*<sup>2</sup>. Se pone en contexto la situación actual, se comentan los dos esquemas de datos y se desarrolla el proceso de migración en dos fases.

Por una parte, se realiza la migración a *OJS* del contenido ya existente en la biblioteca digital, almacenados en *WordPress*, y posteriormente se define la carga anual de los años venideros desde la plataforma de gestión de conferencias *EasyChair*<sup>3</sup>. La figura 1 resume gráficamente el proceso; vemos que existen dos migraciones distintas, en las que aparecen *BDSISTEDES\_WP*, correspondiente al sistema actual en *WordPress*, y *BDSISTEDES\_OJS*, la biblioteca digital en su plataforma destino. El lenguaje utilizado para el intercambio de datos entre plataformas es *XML*.

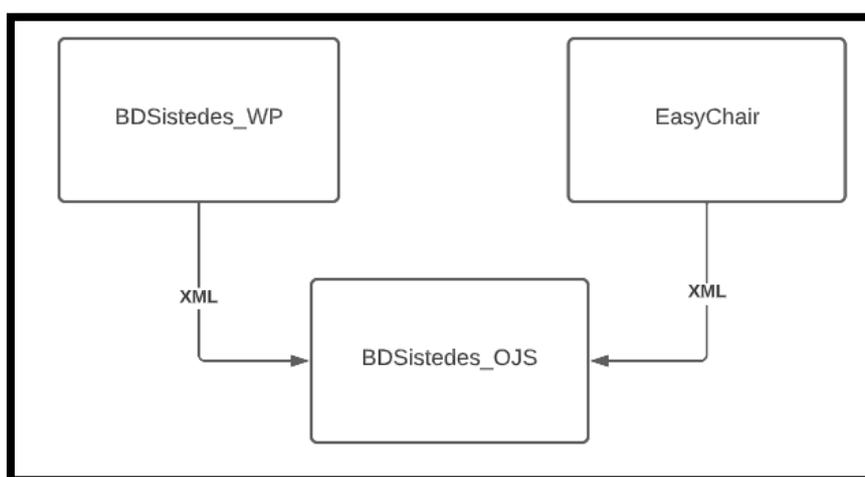


Figura 1: Esquema de migración en BDSISTEDES

<sup>1</sup> <https://wordpress.com/es/>

<sup>2</sup> <https://pkp.sfu.ca/ojs/>

<sup>3</sup> <https://easychair.org/>

## 1.2 Problema

La sociedad ha sufrido un avance tecnológico significativo en muy poco tiempo. Se ha visto cómo ha sido completamente necesaria la adaptación de los recursos de las compañías a nuevas formas de tecnología y negocio.

Este cambio no siempre se produce por un avance tecnológico, sino que en ocasiones es posible que un sistema de gestión necesite actualizarse o encontrar mejores vías más productivas o específicas para un proyecto. En nuestro caso, el cambio se produce para mejorar significativamente la gestión y el almacenamiento de datos de una biblioteca digital.

En estos momentos, se dispone de un sistema que cumple con los requisitos de la biblioteca, pero que no gestiona todas las necesidades de la mejor manera posible, debido a las debilidades que presenta la plataforma *WordPress*, que hacen que se requiera de soluciones *ad-hoc* para algunos aspectos de la gestión del contenido digital.

Este trabajo aporta el diseño y la implementación más adecuados a esta necesidad, mejorando la experiencia de uso de esta. También busca definir una nueva metodología de uso pensando en el futuro.

## 1.3 Solución

El objetivo de este proyecto es diseñar e implementar un proceso de migración para la Biblioteca Digital de SISTEDES.

El destino de esta biblioteca es el software para la administración de revistas *Open Journal Systems (OJS)*. Este software libre aporta una infraestructura técnica que abarca secciones interesantes como el flujo editorial por completo o la indexación.

Conseguir migrar esta biblioteca a *OJS* proporciona una mejora considerable en la gestión y administración de la biblioteca, organizando los artículos por revistas y ofreciendo servicios adicionales que hasta ahora no se tenían.

Además, con este software, es posible pensar en cambios en la biblioteca añadiendo roles para los usuarios. Existen diferentes roles como administrador de revista, editor, revisor, autor, lector, etc.

Cumplir con este objetivo requiere que la siguiente serie de subobjetivos sean cumplidos:

- ❖ Analizar la estructura de *WordPress* y, más concretamente, de *BDSISTEDES*, para gestionar bien la exportación de su contenido.
- ❖ Mapear los datos de entrada almacenados en *WordPress*.
- ❖ Mapear el esquema de metadatos de *BDSISTEDES* al esquema de *OJS*.
- ❖ Analizar la estructura de *OJS* para la implementación de la nueva *BDSISTEDES* y la siguiente importación.
- ❖ Transformar el contenido de acuerdo con el mapeo definitivo.
- ❖ Evitar la pérdida de datos durante la migración.

- ❖ Documentar todo el proceso adecuadamente.

El cambio de software permitirá conseguir una serie de ventajas, incluyendo:

- ❖ Ampliar la indexación del contenido.
- ❖ Estadísticas sobre la actividad, el acceso y las descargas, incluyendo las vistas de artículos y resúmenes.
- ❖ Empleo de la misma estructura para el almacenamiento de los datos, facilitando la búsqueda y la edición.

## 1.4 Estructura

Este documento está dividido en diferentes partes. Primero se hace una introducción al trabajo, seguida de la revisión del contexto tecnológico. Posteriormente, se separan las dos migraciones realizadas en este trabajo: la migración de los datos anteriores a 2022, y la carga anual de este año y las futuras. La migración está definida al detalle con cada paso a seguir y la decisión tomada junto con su justificación.

Se detalla la estructura:

- 1) **Introducción:** se presenta el trabajo a realizar, se habla de la motivación generada para hacerlo, qué objetivos existen y el impacto que se espera conseguir al terminarse.
- 2) **Contexto tecnológico:** se comentan las dos tecnologías involucradas en la migración de datos del proyecto, y qué recursos ofrece el nuevo sistema de gestión de bibliotecas digitales.
- 3) **Arquitectura de la información:** se describen los datos pertenecientes a cada sistema, junto con los de destino.
- 4) **Tecnología utilizada:** se describen las aplicaciones empleadas para la resolución del trabajo.
- 5) **Migración 1:** se desarrolla la migración de *WordPress* a *OJS* paso a paso con todas las consideraciones oportunas de esta solución.
- 6) **Migración 2:** se distingue el nuevo proceso de carga anual de la biblioteca con el año 2022 como prueba efectiva del cambio.
- 7) **Conclusiones:** conclusiones alcanzadas al finalizar el trabajo.
- 8) **Referencias:** referencias bibliográficas del trabajo.



# CAPÍTULO 2 - Contexto tecnológico

---



Una biblioteca digital se suele definir como una colección organizada de objetos digitales, que ofrece servicios a un usuario final, disponiendo de mecanismos de acceso al contenido, así como la conservación de este. El objetivo principal es proporcionar el acceso directo a recursos de manera estructurada y fiable. [Board, 2010]

El desarrollo y mantenimiento de una biblioteca digital necesita una serie de aspectos fundamentales para considerar. Se debe señalar la importancia de los metadatos, ya que permiten obtener búsquedas más efectivas y precisas, a la vez de ayudar a recuperar la información.

A lo largo de las últimas 3 décadas se han llevado a cabo esfuerzos por desarrollar herramientas de gestión y acceso a bibliotecas digitales. Estos esfuerzos han resultado en sistemas software para la gestión de contenido, agrupados algunos de ellos bajo el acrónimo *CMS* (del inglés Content Management System) y otros conocidos como software de repositorio (como *Fedora*<sup>4</sup> y *Dspace*<sup>5</sup>) o, directamente, sistema de biblioteca digital (como *Greenstone*<sup>6</sup>).

Probablemente el universo *CMS* es el que mayor proyección ha tenido en la industria del software, especialmente en la de creación de contenido para la web. Un *CMS* es un software que tiene como objetivo facilitar la gestión, creación y edición de contenidos de una web. Existen muchos tipos de *CMS*, dependiendo del fin al que se quiera llegar, como blogs, foros, comercios electrónicos y bibliotecas digitales. Como ejemplos de *CMS*, se pueden nombrar *Blogger*<sup>7</sup>, *Moodle*<sup>8</sup> y *WordPress*. [TecnoPymes, 2018]

Actualmente, *WordPress* es el *CMS* más usado de todo el mundo. Cerca del 43% de las páginas web que accedemos todos los días están creadas con este software. Normalmente, este sistema es más que suficiente para cualquier propósito, desde un blog personal, hasta una tienda online *e-commerce*.

Nuestro proyecto se centra en la gestión de contenidos en el ámbito académico. Específicamente, vamos a desarrollar un proceso de migración de una biblioteca digital de publicaciones académicas (*BDSISTEDES*) implementada en un *CMS* de propósito general como *WordPress* hacia otro *CMS* más específico, concebido para la gestión de ese tipo de publicaciones.

Existen diferentes *CMS* en este ámbito, como *EPrints*<sup>9</sup>, *Ambra*<sup>10</sup> y *OJS*.

*EPrints* es una aplicación web de software libre y código abierto que se utiliza para la gestión de repositorios digitales en acceso abierto. Es el primer software de repositorio institucional de acceso abierto y uno de los más utilizados. Está basado en la arquitectura *LAMP* (sistema operativo *Linux*, *Apache* como servidor web y *MySQL/MariaDB* para gestionar las bases de datos, y lenguaje *Perl*). [OJS, 2022]

---

<sup>4</sup> <https://getfedora.org/es/>

<sup>5</sup> <https://www.dspace.com/en/pub/home.cfm>

<sup>6</sup> <https://www.greenstone.org/>

<sup>7</sup> <https://www.blogger.com/about/?bpli=1>

<sup>8</sup> <https://moodle.org/?lang=es>

<sup>9</sup> <https://www.eprints.org/uk/>

<sup>10</sup> <https://plos.github.io/ambraproject/>

*Ambra* es otra plataforma de código abierto para la publicación de artículos de investigación. Ofrece características para la discusión en la publicación y una vez está publicado un artículo. El servicio *PLOS* <sup>11</sup>(*Public Library of Science*) ya no desarrolla activamente *Ambra*, aunque sí lo mantiene como su plataforma operativa para el catálogo de revistas de *PLOS*.

*Open Journal Systems (OJS)* es el software elegido para nuestro proyecto. Está activamente en uso en más de 25.000 revistas. En España, el *Consejo Superior de Investigaciones Científicas* utiliza este sistema en su sección de revistas científicas. También, *RACO del Consorcio de Servicio Universitarios de Cataluña (CSUC)* publica alrededor de 500 revistas con esta herramienta. Muchas universidades españolas también han optado por este software para la gestión de sus artículos, como la *Universidad Autónoma de Barcelona*, la *Universidad Complutense de Madrid* o la *Universidad de Valencia*. Se emplea además en muchos otros países, siendo ejemplos la *Biblioteca Nacional de Perú*, universidades de Venezuela, *Universidad Nacional de La Plata* en Argentina y muchos más lugares. Es por ello por lo que OJS va a ser la plataforma destino en nuestro proceso de migración.

A continuación, vamos a describir con un poco más de detalle las plataformas origen y destino.

## 2.1 WordPress

*WordPress* es un sistema de gestión de contenidos basado en *PHP*, *MySQL* y con licencia *GPL*<sup>12</sup>.



Figura 2: Logo oficial de WordPress

---

<sup>11</sup> <https://plos.org/>

<sup>12</sup> [https://es.wikipedia.org/wiki/GNU\\_General\\_Public\\_License](https://es.wikipedia.org/wiki/GNU_General_Public_License)

*WordPress* surgió como una plataforma del tipo *blog*, en la que se mostraban las entradas ordenadas por fecha. Posteriormente, se añadió la posibilidad de construir también sitios basados en páginas estáticas. El diseño visual de un sitio se proporciona a través de plantillas, con la posibilidad de elegir entre miles de ellas, ya sean de pago o gratuitas. En la mayoría de las plantillas se separa el contenido y el diseño *HTML* y *CSS*. Existen bloques que sirven para diferentes funciones elegidas por el usuario, a través de widgets. [Wikipedia, 2022]

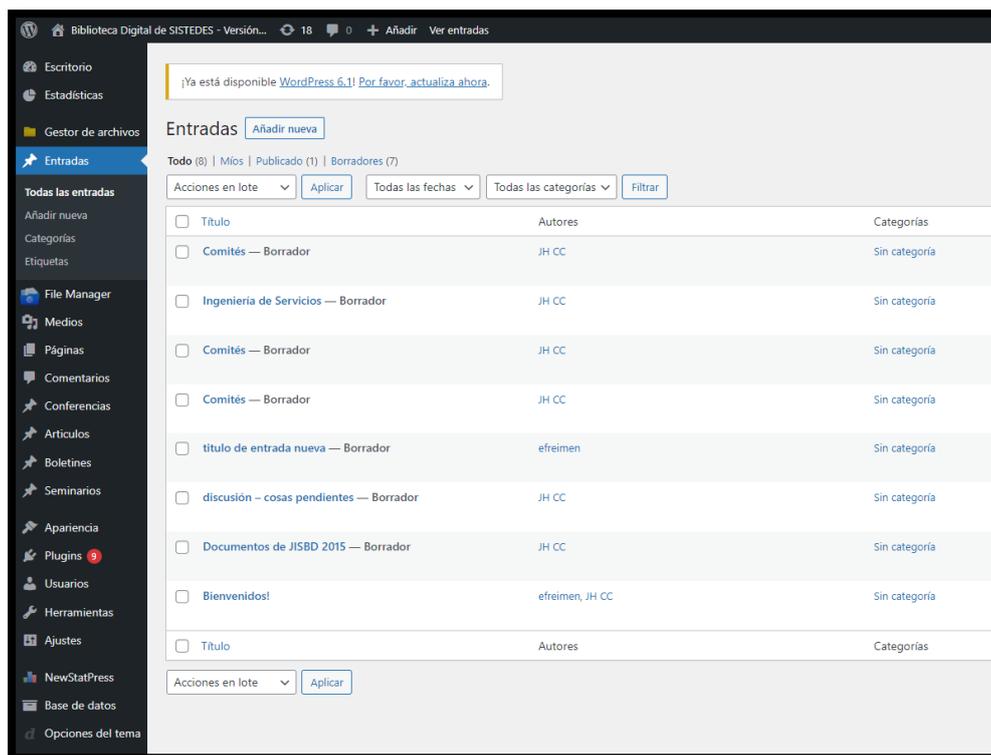


Figura 3: Interfaz WordPress

*WordPress* tiene un mecanismo de extensibilidad basado en la idea complemento o *plugin*, permitiendo expandir la funcionalidad por defecto del software. Es posible utilizar un *plugin* cargado en línea, o subir uno directamente desde nuestro ordenador.

Otras características de *WordPress* son las siguientes:

- ❖ Fácil instalación, actualización y personalización.
- ❖ Sistema de usuarios o autores, con diferentes roles o perfiles con niveles de permisos.
- ❖ Estados por entrada: publicado, borrador, esperando revisión, privado, protegido con contraseña.
- ❖ Publicación mediante correo electrónico.
- ❖ Guardado automático temporizado del artículo como borrador.
- ❖ Gestión y distribución de enlaces.
- ❖ Subida y gestión de datos adjuntos y archivos multimedia.

La actual *BDSISTEDES* está implementada sobre *WordPress*. Aunque no es un sistema específico para este fin, la funcionalidad de base se completó con un *plugin* que implementa algunos aspectos, mientras que otros *plugins* externos aportaron el resto. Usando las capacidades de exportación de *WordPress*, se generará un conjunto de metadatos que deberá ser migrado al nuevo sistema.

## 2.2 *EasyChair*

*EasyChair* es un sistema de administración de conferencias, hospedado por el departamento de ciencias de la computación de la Universidad de Manchester. Es utilizado en la gestión de las contribuciones a las conferencias, automatizando el envío, recepción y otras acciones relacionadas. La versión de pago cuenta con servicio web, número limitado de almacenamiento y soporte técnico.

*EasyChair* ofrece un sistema de exportación que permite descargar los metadatos de los artículos aceptados para su presentación en las conferencias, pudiendo separar los autores, los documentos y los metadatos de los mismos. [EasyChair, 2022]

SISTEDES utiliza esta plataforma para la gestión de sus conferencias. Hasta ahora, los metadatos se volcaban a *WordPress* y todo el contenido quedaba guardado ahí. Después de la migración, el proceso será distinto, pues estos datos serán volcados a *OJS*.



Figura 4: Logo oficial de EasyChair

## 2.3 *Open Journal System (OJS)*

En este proyecto hacemos uso de *OJS* como plataforma destino de la migración desde *BDSISTEDES*. Por ello, se van a comentar las características de esta herramienta y los aspectos importantes para tener en cuenta a la hora de trabajar con ella.

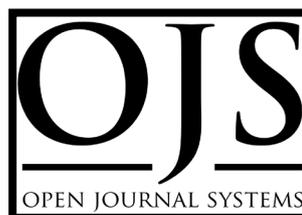


Figura 5: Logo oficial de OJS

*OJS* es un software libre para la gestión de revistas desarrollado por el Public Knowledge Project, una iniciativa de investigación sin ánimo de lucro formada por diferentes universidades, y fundada en 1998. Está liberado bajo licencia GNU General Public License, garantizando la libertad de usar, estudiar, compartir y modificar el software. [OJS, 2022]

El objetivo de *OJS* es mejorar la calidad de la publicación de revistas a través de innovaciones como mejorar la experiencia del lector, la transferencia de las políticas de las revistas y la mejora de la indización. Cubre todos los aspectos de la publicación de revistas en línea, desde el establecimiento de un sitio web hasta las tareas operativas tales como el proceso de envío de autor, revisión por pares, la edición, la publicación, el archivo y la indexación de la revista. Ayuda también en el manejo de aspectos personales de la organización de una revista, incluyendo el seguimiento del trabajo de editores, revisores y autores, notificando a los lectores y ayudando con la correspondencia.

### 2.3.1 Ejemplos de sitios con OJS

Actualmente es posible encontrar muchos ejemplos de páginas web que emplean este software libre. En este apartado se van a comentar dos ejemplos.

#### Universidad de Málaga (UMA)

La Editorial de la Universidad de Málaga dispone de un apartado en su página web donde almacena una serie de revistas científicas sobre distintos temas. La figura 5 muestra la página de entrada a una de las revistas alojadas. [UMA, 2022]

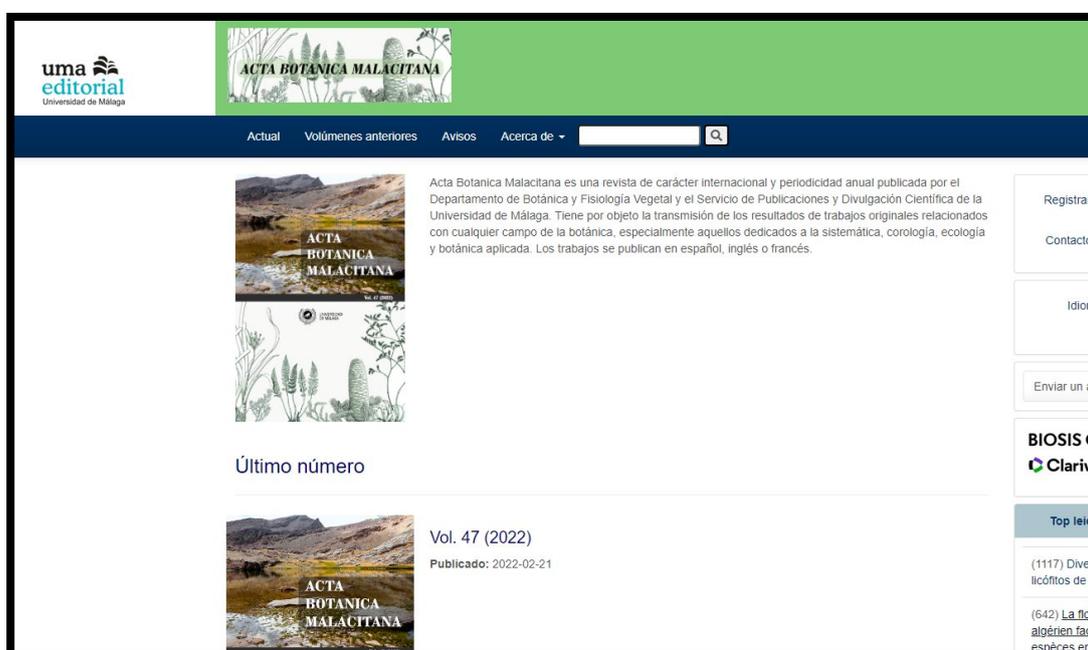


Figura 6: Revista UMA

En general, se puede escoger el número actual o seleccionar uno de los anteriores. La vista de un artículo contiene el resumen, referencias del mismo y artículos más leídos del

mismo autor (ver Figura 7). Cada artículo contiene su documento *PDF* disponible para visualizar.



Figura 7: Artículo en revista UMA

### Universitat Politècnica de València (UPV)

La *UPV* dispone de un portal que impulsa el empleo de las revistas científicas digitales a través de *OJS*. [UPV, 2022]

Dispone de una herramienta para solicitar publicar una revista. Mediante correo o desde un formulario web permiten solicitar alojamiento en la plataforma, exigiendo una serie de requisitos.



Figura 8: Portal de revistas de la UPV

Añade un apartado de los artículos más populares del mes anterior, con el número de visitas. Además, posee un apartado de guía en el que explica el funcionamiento de la biblioteca, sobre aspectos como el registro e inicio de sesión, el envío de un artículo o el flujo de los revisores.

### Universidad de Valencia (UV)

La *Universidad de Valencia* pone a disposición de la comunidad académica una plataforma en *OJS* para la gestión editorial de revistas científicas electrónicas. La plataforma, cuya página de entrada aparece en la Figura 9, ofrece facilidades para la gestión de los procesos implicados en la producción de una revista, desde la recepción de propuestas de artículos por los investigadores hasta la publicación electrónica en Internet. [UV, 2022]

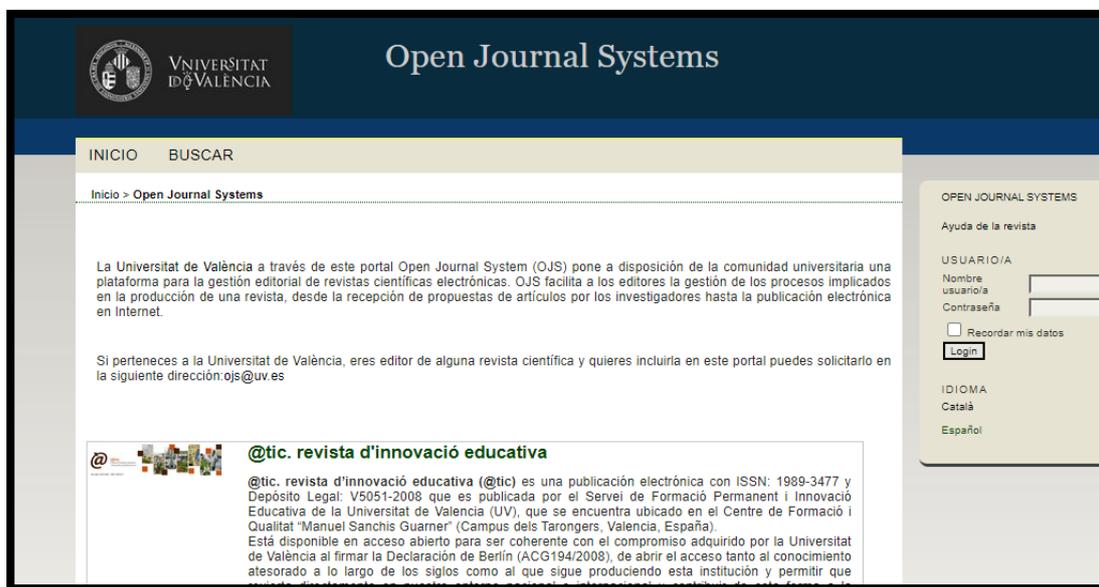


Figura 9: Portal de revistas de la UV

La figura 9 muestra la vista de un número de *Celestinesca*, una revista de esta biblioteca. Es interesante observar que aparece un menú en la parte superior con la información de ese número y el acceso a los volúmenes que contenga y sus correspondientes artículos.



Figura 10: Número de una revista UV

## 2.4 Propuesta

La biblioteca digital de SISTEDES ha utilizado desde 2015 *WordPress* como software para la gestión de su contenido. Este sistema presenta una serie de limitaciones que han llevado a la decisión de hacer este trabajo.

Por una parte, existen vulnerabilidades que afectan a la seguridad de la biblioteca. La falta de actualización en temas o *plugins* puede ser una causa de estas vulnerabilidades. Al trabajar entre diferentes versiones, se crean incompatibilidades que pueden producir errores, provocando pérdida de datos.

También, la falta de adaptación al dominio de trabajo anima a cambiar de sistema de gestión. Al ser una biblioteca digital, se necesita un proceso editorial específico y ciertas herramientas de gestión de colecciones que un software tan amplio como *WordPress* no puede ofrecer.

Este trabajo conseguirá que la existente biblioteca gestionada a través de *WordPress*, se pueda gestionar desde *OJS*. Gracias a la migración de sistema se conseguirá una mejora en la gestión de la biblioteca, empleando un sistema específico.

# CAPÍTULO 3 - Arquitectura de la información

---



En este trabajo, abordamos un problema de migración de datos desde un esquema a otro diferente. Dicha migración, como ya se ha comentado, se realizará en dos fases, una para el contenido ya existente en *BDSISTEDES*, y otra para aplicar anualmente a partir de *EasyChair*.

En ambos casos, el principal problema a abordar en el proceso de migración es la heterogeneidad semántica entre el esquema de metadatos de origen y destino. La definición de las transformaciones entre esquemas va a ser, en consecuencia, una de las tareas fundamentales de nuestro trabajo. Como primer paso, en este capítulo ofrecemos detalles sobre dichos esquemas tanto en *BDSISTEDES* como en *EasyChair* y *OJS*.

### 3.1 Origen 1: *BDSISTEDES*

*BDSISTEDES* aloja diferentes tipos de contenido digital, generado por los miembros de la comunidad en diferentes manifestaciones:

- ❖ **Seminarios:** charlas sobre temas de interés para la comunidad científica, impartidas por destacados miembros de esta y/o reconocidos expertos en las materias objeto de los mismos. En *BDSISTEDES* se guarda, junto a los metadatos básicos, una grabación en vídeo del seminario.
- ❖ **Boletines:** publicaciones bimensuales que recopilan noticias recientes y relevantes, tanto para los socios como para todo aquel que pueda estar interesado en la asociación. Son el tipo de contenido más sencillo, pues básicamente se almacena su contenido en un documento *PDF*.
- ❖ **Actas de Conferencias:** almacenan los trabajos presentados por los investigadores en tres foros de ámbitos relacionados con la Ingeniería del Software. Estos foros son:
  - *PROLE*: Jornadas sobre Programación y Lenguajes.
  - *JISBD*: Jornadas de Ingeniería del Software y Bases de Datos.
  - *JCIS*: Jornadas de Ciencia e Ingeniería de Servicios.

La figura 11 muestra una captura de la jerarquía de publicaciones, tal y como se muestra en la página principal de *BDSISTEDES*. Los documentos *SISTEDES* corresponden con un apartado todavía no incluido en la página. Las conferencias a su vez están organizadas por ediciones, siendo la primera disponible la de 2015.



Figura 11: Interfaz en BDSISTEDES



Figura 12: Tracks en BDSISTEDES

Cada conferencia está dividida en ediciones anuales, cada una de las cuales a su vez se estructura en secciones temáticas o *tracks* (ver Figura 12).

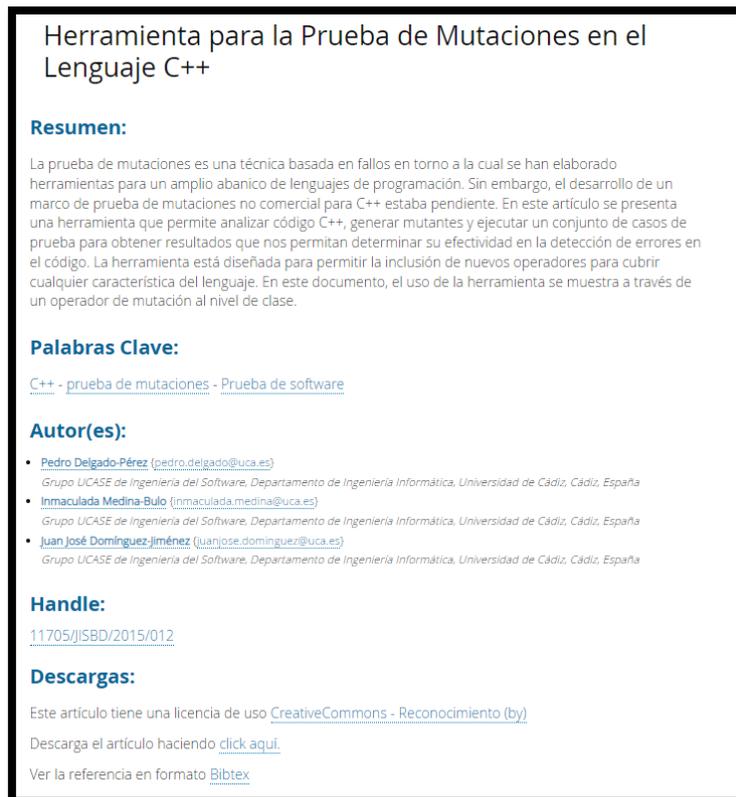


Figura 13: Artículo en BDSISTEDES

Por último, se muestra en la Figura 13 una vista general de un artículo en *WordPress*. Se puede ver que encontramos un título, un resumen, un apartado de palabras clave, los autores del artículo, el identificador *handle* y un apartado para descargar el contenido.

## 3.2 Origen 2: *EasyChair*

*EasyChair* ofrece una herramienta de exportación que permite obtener los metadatos de la información almacenada en documentos separados. La Figura 14 muestra los archivos generados en la exportación de los metadatos correspondientes a la conferencia *JISBD 2022*.

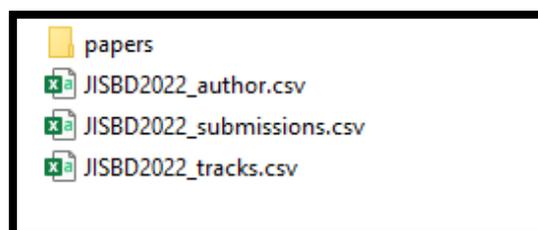


Figura 14: Exportación de *EasyChair*

Como vemos, se obtiene una carpeta *papers*, que contiene los artículos como archivos PDF. Además, se generan tres documentos CSV:

- ❖ *JISBD2022\_author.csv* : contiene los metadatos de los autores los artículos de esta conferencia.
- ❖ *JISBD2022\_submissions.csv* : contiene los metadatos de los artículos de esta conferencia.
- ❖ *JISBD2022\_tracks.csv* : contiene los metadatos de los *tracks* de esta conferencia.

Los elementos que aparecen en el documento *JISBD2022\_submissions.csv*, que es el más relevante para nosotros, se pueden ver en el esquema mostrado en la Figura 15.

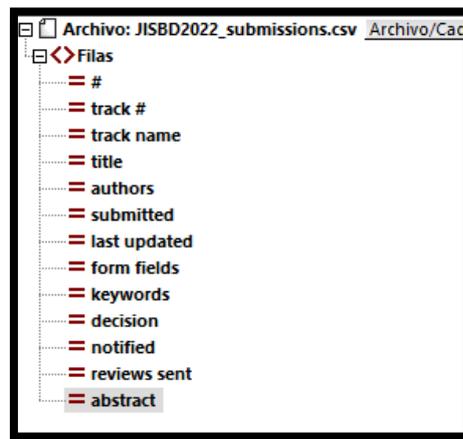


Figura 15: Fichero *submissions.csv* EasyChair

### 3.3 Destino: *BDSISTEDES* en *OJS*

La arquitectura en la que se basa la solución de este trabajo se muestra de manera visual a través del esquema de la figura 17. Como se observa, se van a crear cinco revistas, correspondientes a seminarios, boletines, Jornadas de Ingeniería del Software y Bases de Datos (*JISBD*), Jornadas sobre Programación y Lenguajes (*PROLE*) y las Jornadas de Ciencia e Ingeniería de Servicios (*JCIS*).

Los seminarios y los boletines se representarán como números o *issues* de la correspondiente revista; cada *issue*, posee su identificador y además una parte de secciones que se utilizan para dividir los artículos según los intereses de cada revista.

Las actas de conferencia (*JCIS*, *PROLE*, *JISBD*) se dividen en ediciones por año, correspondiendo cada edición a una *issue* en la revista correspondiente. Los artículos presentados en los diferentes *tracks* se agrupan en conjuntos de artículos dentro de un *issue*.

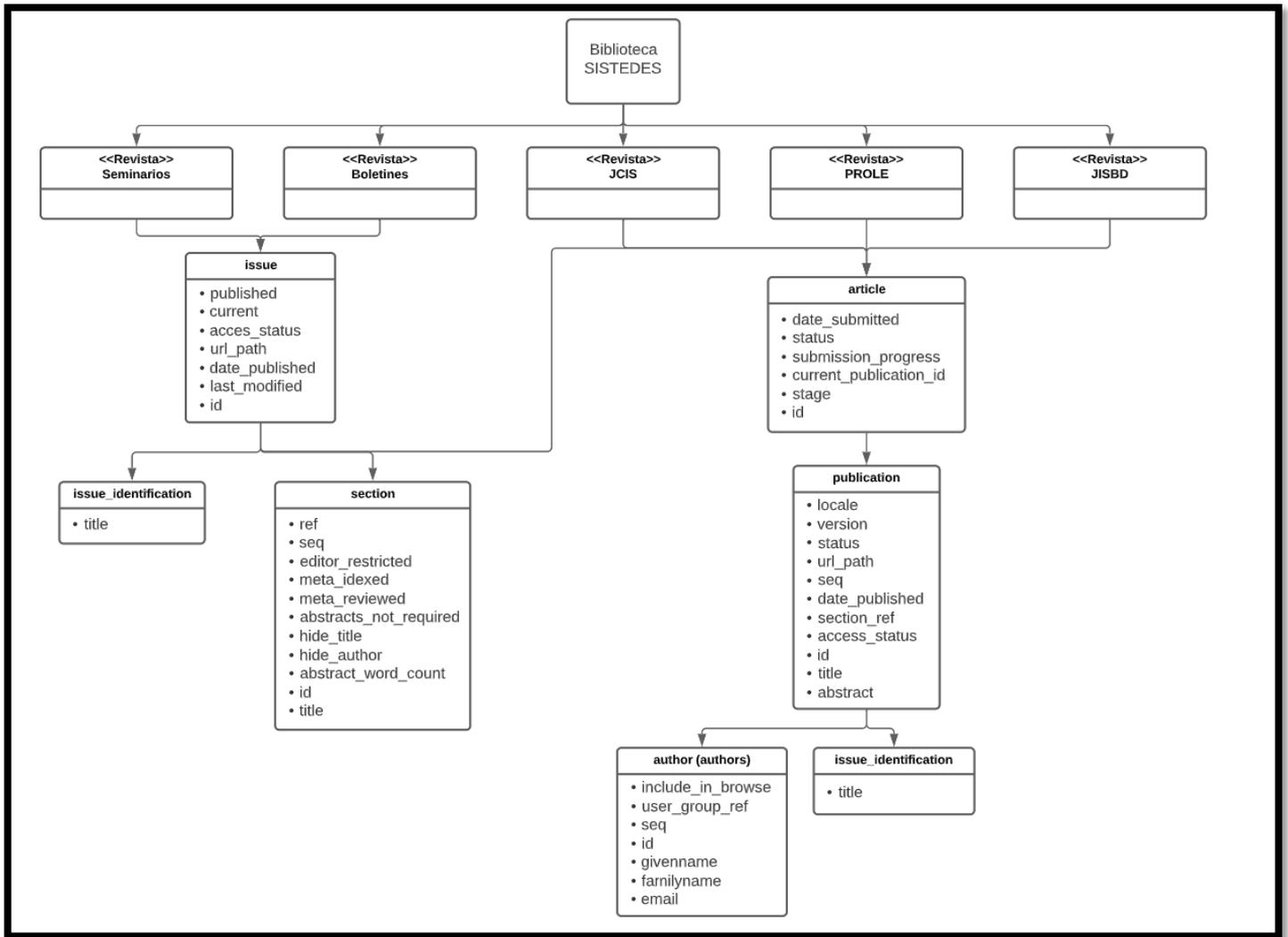


Figura 16: Estructura de BDSISTEDES en OJS

En las actas de conferencias (*JCIS*, *JISBD* y *PROLE*) aparecen las diferentes ediciones representadas como *issues*, como se puede ver en la figura 17.

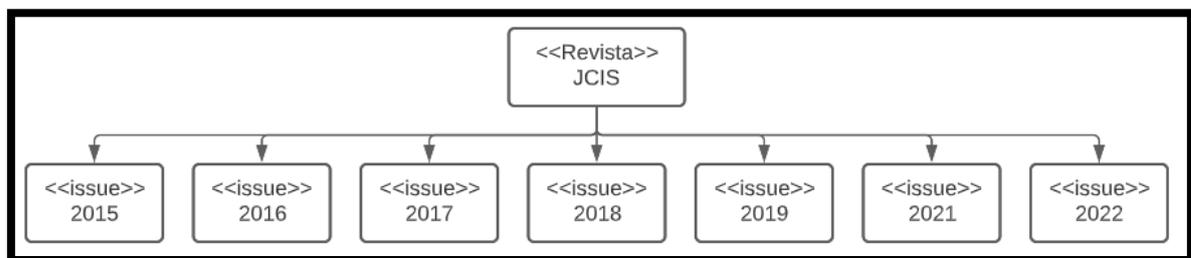


Figura 17: Estructura de las actas de conferencias en OJS

OJS ofrece diferentes formatos de importación a la hora de migrar datos, dependiendo de qué estructura se pretende crear en la plataforma.

```

<id type="internal" advice="ignore"><!-- Next ID from OJS or blank --></id>
<title locale="en_US"><!-- Issue title in English or blank --></title>
<title locale="uk_UA"><!-- Article title in Ukrainian or other non-English
<abstract locale="en_US"><!-- Abstract in English or blank --></abstract>
<abstract locale="uk_UA"><!-- Abstract in Ukrainian or other non-English
<copyrightHolder locale="en_US"><!-- Copyright Holder in English or blank
<copyrightHolder locale="uk_UA"><!-- Copyright Holder in Ukrainian or othe
<copyrightYear><!-- Copyright Year --></copyrightYear>
<keywords locale="en_US">
  <keyword><!-- Article Keyword in English --></keyword>
  <keyword><!-- Article Keyword in English --></keyword>
</keywords>
<keywords locale="uk_UA">
  <keyword><!-- Keyword in Ukrainian or other non-English languages -->
  <keyword><!-- Keyword in Ukrainian or other non-English languages -->
</keywords>
<authors xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schema
  <author include_in_browse="true" user_group_ref="Author">
<!-- user_group_ref= User group in your system for this Author -->
  <firstname><!-- First name or Name and patronymic --></firstname>
  <lastname><!-- Last name --></lastname>
  <affiliation locale="en_US"><!-- Affiliation in English --></affiliati
  <affiliation locale="uk_UA"><!-- Affiliation in Ukrainian or other non
  <country><!-- Country code, for example: UA --></country>
  <email><!-- Author Email --></email>
  <orcid><!-- ORCID full URL, for example: https://orcid.org/XXXX-XXXX-X
  <biography locale="en_US"><!-- About Author or Supervisor Additional I
  <biography locale="uk_UA"><!-- About Author or Supervisor Additional I
  </author>
</authors>
<issue_identification>
  <number><!-- Issue number, for example: 1 --></number>
  <year><!-- Year of issues publication --></year>
  <title locale="en_US"><!-- Issue title in English --></title>
  <title locale="uk_UA"><!-- Issue title in Ukrainian or other non-English

```

Figura 18: XML destino OJS

La Figura 18 muestra un ejemplo de documento XML de carga de artículos. Por una parte aparece *submission\_file*, que corresponde con el documento a cargar, codificado. Después aparece *publication*, que contiene los metadatos del artículo a mostrar. Hay bastantes metadatos que coinciden en ambos esquemas, pero también aparecen diferencias en formatos.

# CAPÍTULO 4 - Tecnología utilizada

---



Para el desarrollo de este trabajo, como se ha comentado anteriormente, se han utilizado diferentes herramientas, enfocadas principalmente en el trabajo con documentos *XML*. Se asume el uso del software *OJS*, ya comentado anteriormente.

Se han empleado dos programas dedicados al trabajo con *XML* creados por *Altova* [Altova, 2022]:

- ❖ *XMLSpy*<sup>13</sup>: entorno de desarrollo y editor *XML*, para modelar, editar, transformar y depurar
- ❖ *MapForce*<sup>14</sup>: herramienta gráfica que permite asignar, convertir e integrar cualquier combinación de datos, y transformarlos para el resultado que se desee

Para conseguir la codificación de los documentos *PDF* (*papers*) de cada artículo se ha ejecutado un programa *PHP* que accede a la dirección de cada artículo y se guarda el propio documento codificado.

Para la edición y la visualización de ficheros *XML* también se ha empleado *Visual Studio Code*<sup>15</sup>, una herramienta cómoda compatible con muchos lenguajes y con opciones cómodas de edición múltiple, bastante útil en algunos puntos del trabajo.

## 4.1 Altova XMLSpy

Es el editor de *XML* y *JSON* más vendido del mundo para modelar, editar transformar y depurar *XML* y otras tecnologías relacionadas.

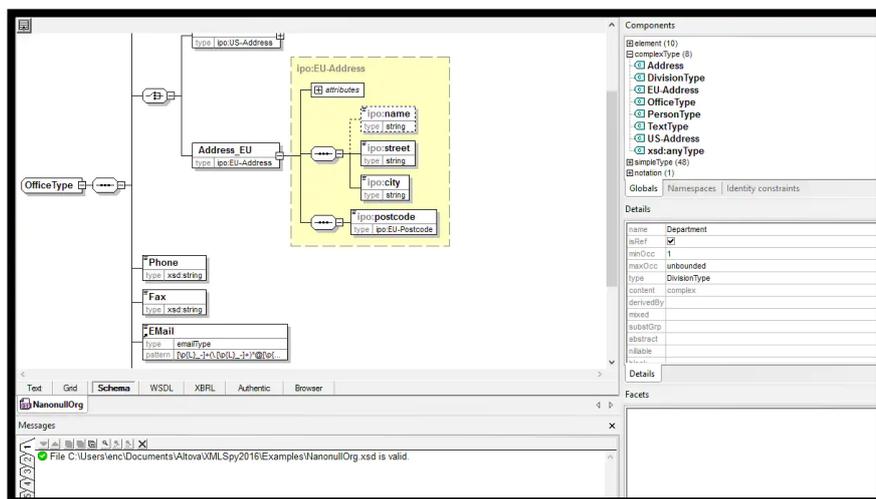


Figura 19: Editor XMLSpy de Altova

<sup>13</sup> <https://www.altova.com/es/xmlspy-xml-editor>

<sup>14</sup> <https://www.altova.com/es/mapforce>

<sup>15</sup> <https://code.visualstudio.com/>

Ofrece todas las herramientas necesarias para crear aplicaciones sofisticadas, como un diseñador de esquemas gráfico, funciones de generación de código, conversores de archivos, etc.

Cuenta con la tecnología *RaptorXMLr*<sup>16</sup> para validar y procesar *XML* y *JSON* datos a alta velocidad. También está disponible como producto servidor multiplataforma. Esta herramienta incluye vistas de edición gráficas, generación de código, asistentes y otras ayudas inteligentes de edición. Características destacadas de esta aplicación pueden ser las siguientes:

- ❖ Editor de *JSON* y editor de *JSON Schema*
- ❖ Transformación de datos *JSON* con *Xpath*, *XSLT*, y *Xquery*
- ❖ Edición de documentos *XML*
- ❖ Función de validación *XML* con corrección automática de errores
- ❖ Editor de esquemas *XML*
- ❖ Editor *XSLT*
- ❖ Generación de código *Java*, *C#* y *C++*
- ❖ *Editor Xquery*

A medida que se trabaja, *XMLSpy* usa la función de validación *XML* inteligente, un proceso que detecta errores y sugiere posibles soluciones con un clic. Corrige el error automáticamente.

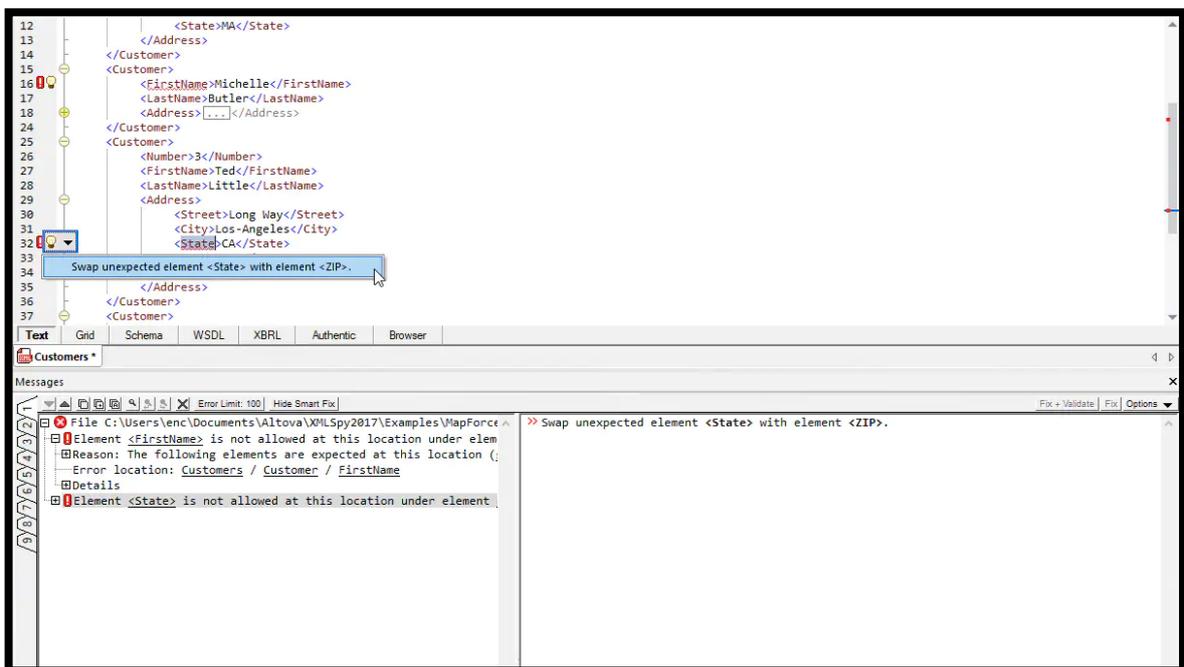


Figura 20: Corrección de errores con *XMLSpy*

<sup>16</sup> <https://www.altova.com/es/raptorxml>

Gracias a esta herramienta, ha resultado sencilla la visualización y edición de los documentos necesarios para hacer la migración. Se han corregido muchos errores de importación y ha ayudado a la comprobación de los datos.

## 4.2 Altova MapForce

Altova *MapForce* es una herramienta gráfica que permite convertir e integrar cualquier combinación de datos. En este trabajo esta herramienta ha sido indispensable para el éxito del mismo, pues la primera parte de la migración empieza con el mapeado de los metadatos de *WordPress* a *OJS*.

Las herramientas de asignación de datos de *MapForce* son las siguientes:

- ❖ Diseño gráfico de asignaciones de datos
- ❖ Integración de datos
- ❖ Funciones de procesamiento de datos
- ❖ Depurador interactivo de asignaciones de datos
- ❖ ETL para empresas
- ❖ Proyectos de asignación entre todo tipo de datos
- ❖ Transformaciones de asignaciones de datos en cadena
- ❖ Genera archivos de ejecución y documentación para las asignaciones

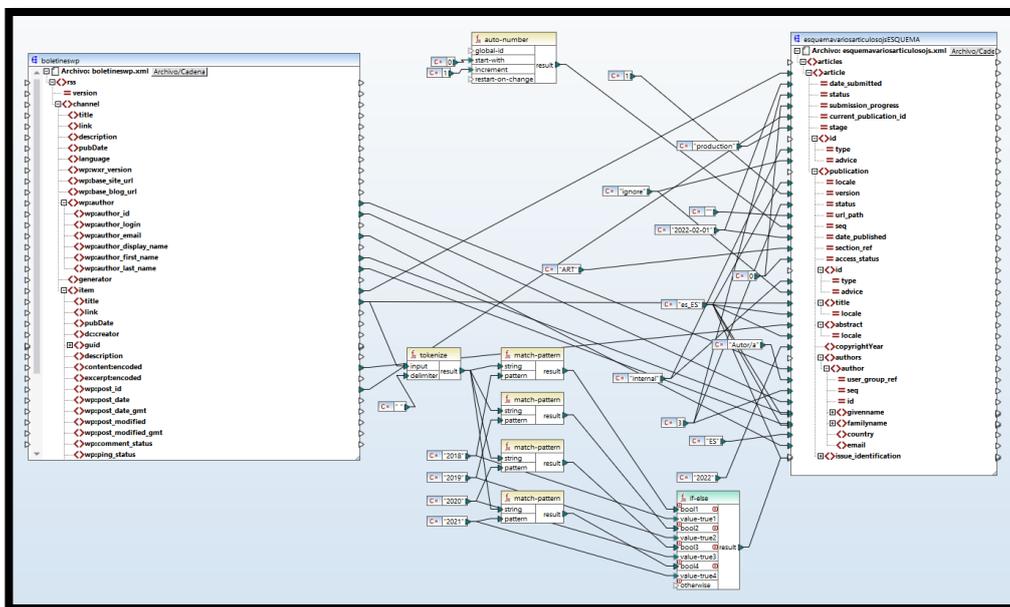


Figura 21: Diseño de transformación en MapForce

## 4.3 *Visual Studio Code*

Es un editor de código fuente desarrollado por *Microsoft* que incluye soporte para la depuración, control integral de *Git*, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. [Visual Studio, 2022]

Se basa en *Electron*<sup>17</sup>, un *framework* que se utiliza para implementar *Chromium* y *Node.js* como aplicaciones para escritorio, que se ejecuta en el motor de diseño *Blink*. Es compatible con varios lenguajes de programación y con un conjunto de características que pueden o no estar disponibles para un lenguaje dado. Permite cambiar la página de códigos en la que se guarda el documento activo, el carácter que identifica el salto de línea y el lenguaje de programación del documento activo.

Una vez comentada la propuesta del trabajo, el impacto esperado, y el diseño de la solución, en este apartado se va a detallar cómo se ha llegado a la solución final. El desarrollo que ha tenido el trabajo separando por secciones cada problema con su solución.

---

<sup>17</sup> <https://www.electronjs.org/>

# CAPÍTULO 5 - Migración 1: Contenido legado

---



En este capítulo se comenta la migración de los metadatos legados, es decir, los datos correspondientes al contenido actual de *BDSISTEDES*. Estos datos almacenan las actas de conferencias, seminarios y boletines desde el año 2015.

El diseño de migración de la web se descompone en diferentes fases, siguiendo buenas prácticas de migración con sistemas de seguridad y división por lotes. Para esta migración se ha empleado *XML* como lenguaje de intercambio, dado que *WordPress* y *OJS* lo emplean en sus procesos de exportación e importación, respectivamente.

Para evitar caídas accidentales del servicio de *BDSISTEDES*, hemos replicado el repositorio *WordPress* original para hacer pruebas y modificaciones, manteniendo el repositorio original intacto y en funcionamiento. Lo mismo con el destino, se ha hecho una instalación en local de *OJS* y se ha trabajado con el formato de importación correcto y las posibilidades de la herramienta.

Se va a detallar cómo se ha llegado a la solución final, explicando las decisiones tomadas en la definición de las transformaciones.

## 5.1 Descripción de la fuente de datos

Si se analiza el código resultante de la descarga, se observa que está organizado con una estructura *key-value*, donde para todo metadato existe una clave. Por ejemplo, las palabras clave o *keywords* aparecen con el *meta\_key* *keywords*, y el *meta\_value* son las palabras clave de ese artículo.

```
<wp:meta_key><![CDATA[paper_pdf]]></wp:meta_key>
<wp:meta_value><![CDATA[descargas/2018/3ISBD/2018-3ISBD-034.pdf]]></wp:meta
</wp:postmeta>
  <wp:postmeta>
    <wp:meta_key><![CDATA[paper_pdf_file]]></wp:meta_key>
    <wp:meta_value><![CDATA[descargas/2018/3ISBD/2018-3ISBD-034.pdf]]></wp:meta
    </wp:postmeta>
      <wp:postmeta>
        <wp:meta_key><![CDATA[author_name_1]]></wp:meta_key>
        <wp:meta_value><![CDATA[Carmen L. Carvajal]]></wp:meta_value>
        </wp:postmeta>
          <wp:postmeta>
            <wp:meta_key><![CDATA[author_email_1]]></wp:meta_key>
            <wp:meta_value><![CDATA[carmen.carvajal07@gmail.com]]></wp:meta_value>
            </wp:postmeta>
              <wp:postmeta>
                <wp:meta_key><![CDATA[author_univ_1]]></wp:meta_key>
                <wp:meta_value><![CDATA[Universidad Politécnica de Madrid - Spain]]></wp:me
                </wp:postmeta>
                  <wp:postmeta>
                    <wp:meta_key><![CDATA[author_name_2]]></wp:meta_key>
                    <wp:meta_value><![CDATA[Ana M. Moreno]]></wp:meta_value>
                    </wp:postmeta>
                      <wp:postmeta>
                        <wp:meta_key><![CDATA[author_email_2]]></wp:meta_key>
                        <wp:meta_value><![CDATA[ammoreno@fi.upm.es]]></wp:meta_value>
                        </wp:postmeta>
                          <wp:postmeta>
                            <wp:meta_key><![CDATA[author_univ_2]]></wp:meta_key>
                            <wp:meta_value><![CDATA[Universidad Politécnica de Madrid - Spain]]></wp:me
```

Figura 22: Metadatos de *BDSISTEDES*

Además de *keywords*, aparecen el resto de metadatos correspondientes al artículo, como resumen, autores, identificador o handle o título. Los artículos suelen tener un

documento *PDF* asociado al mismo, donde aparece el contenido completo del artículo. Para la obtención de los ficheros *PDF* de cada artículo, hay que centrarse en la ubicación de cada uno en la base de datos. Desde 2017, existe un patrón que permitirá automatizar el proceso de obtención de cada uno de los *PDF* necesarios. Durante los años 2015 y 2016, no se tuvo en cuenta una posible migración y no hay ningún patrón encontrado para poder informatizar el proceso de extracción. No obstante, el resto de los datos sí que se han conseguido recuperar.

Los seminarios y los boletines se descargan en ficheros separados. Los metadatos que los componen se resumen en un título y una descripción, que dentro contiene el link del identificador y el enlace de descarga.

```

<title>
  <![CDATA[ Boletín nº 2. Abril de 2018 ]]>
</title>
<link>https://bdsdev.dsic.upv.es/boletin/boletines-de-prensa/2018-2/boleti
<pubDate>Wed, 02 May 2018 07:19:07 +0000</pubDate>
<dc:creator>
  <![CDATA[ jhcanos ]]>
</dc:creator>
<guid isPermaLink="false">https://biblioteca.sistedes.es/?post_type=boleti
<description/>
<content:encoded>
  <![CDATA[ Boletín SISTEDES abril de 2018. Handle: <a href="http://hdl.ha
</content:encoded>
<excerpt:encoded>
  <![CDATA[ ]]>
</excerpt:encoded>
<wp:post_id>2839</wp:post_id>
<wp:post_date>
  <![CDATA[ 2018-05-02 09:19:07 ]]>
</wp:post_date>
<wp:post_date_gmt>
  <![CDATA[ 2018-05-02 07:19:07 ]]>
</wp:post_date_gmt>
<wp:post_modified>
  <![CDATA[ 2018-05-04 08:42:43 ]]>
</wp:post_modified>
<wp:post_modified_gmt>
  <![CDATA[ 2018-05-04 06:42:43 ]]>
</wp:post_modified_gmt>
<wp:comment_status>
  <![CDATA[ closed ]]>
</wp:comment_status>
<wp:ping_status>
  <![CDATA[ closed ]]>
</wp:ping_status>
<wp:post_name>

```

Figura 23: Metadatos de los boletines de BDSISTEDES

## 5.2 Definición de la transformación

### 5.2.1 Handles

*BDSISTEDES* utiliza el sistema *Handle* para gestionar la identificación única de los artículos que almacena. Es una infraestructura distribuida que permite localizar a objetos digitales vía una cadena única, independientemente de su localización <sup>18</sup>. [Wikipedia, Handle, 2022]

<sup>18</sup> <https://www.handle.net>

Un handle es una cadena alfanumérica que se compone de dos partes principales: la autoridad global, y la autoridad local (ver Figura 20).

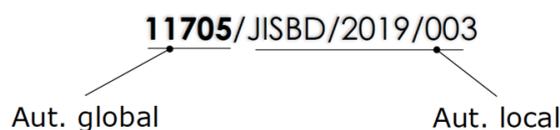


Figura 24: Estructura de un handle

La autoridad global es el primer código numérico, asignado por el sistema y único en toda la infraestructura. En el caso de SISTEDES, dicho código es 11705, como se ve en la Figura 20. El resto del identificador corresponde a la llamada autoridad local, y es gestionado por la propia biblioteca digital para asegurar que no existen dos objetos digitales con el mismo valor. De ese modo, se garantiza la unicidad a nivel global de los identificadores. *BDSISTEDES* ha escogido un modelo jerárquico para la creación de autoridades locales, de modo que en general el patrón seguido es el siguiente:

- ❖ Para conferencias: 11705/{JISBD|PROLE|JCIS}/AÑO/ID (ver Figura 20).
- ❖ Para boletines: 11705/BOLETIN/AÑO/NÚMERO .
- ❖ Para Seminarios: 11705/SEM/SEMnnn, donde nnn es el número de seminario.

Este dato se ha asignado como identificador de artículo de tipo *URN*, porque se administran sobre el *plugin URN*. Se ha escogido así ya que permite respetar la estructura que siguen los identificadores hasta ahora.

Este módulo activa la asignación del *URN* (Uniform Resource Names) de los números, artículos y galeradas en *OJS*. Las galeradas (en inglés *galleys*) contienen el contenido de un artículo, normalmente en formato *PDF*.

Es muy interesante pensar en este *plugin* en relación a la biblioteca de SISTEDES, ya que permite editar de forma sencilla la estructura que se desee asignar a los identificadores. Por ejemplo, se puede decidir la dirección *URL* de resolución, lo que permite asignar un identificador desde la página que se desee.

Como la dirección a la que apuntan los *handles* va a ser diferente cuando termine la migración, es necesario obtener un documento que guarde las nuevas direcciones de los artículos. Para ello, se observa la *URL* final del mismo. Un ejemplo de la dirección es la siguiente:

<http://SISTEDES.es/JCIS/article/view/10889>

El número que aparece corresponde con el identificador del artículo que se está viendo, por lo que se puede relacionar exportando los datos de *OJS*. Al obtener el fichero *XML* con los metadatos de los artículos, aparece ese identificador que se muestra en la dirección, por lo que es posible conseguir una tabla con todos los *handles* conectados con la nueva dirección.

### 5.2.2 Secciones

*OJS* dispone de secciones que permiten dividir los artículos de un issue. Por defecto todos los artículos se guardarán en la sección “*Artículos*”. Es posible crear las secciones que se deseen desde las opciones de administración o directamente al cargar un nuevo issue. Se necesita añadir una abreviatura que servirá de referencia a la misma.

Dentro de las opciones se puede elegir entre distintas funciones como desactivar la sección, quitar la obligación de que aparezca un resumen, no incluirse en el índice de la revista, etc. La web de *SISTEDES.es* está configurada de forma que dentro de cada año de publicaciones existen diferentes *tracks*, agrupando artículos según unos criterios comunes entre ellos.

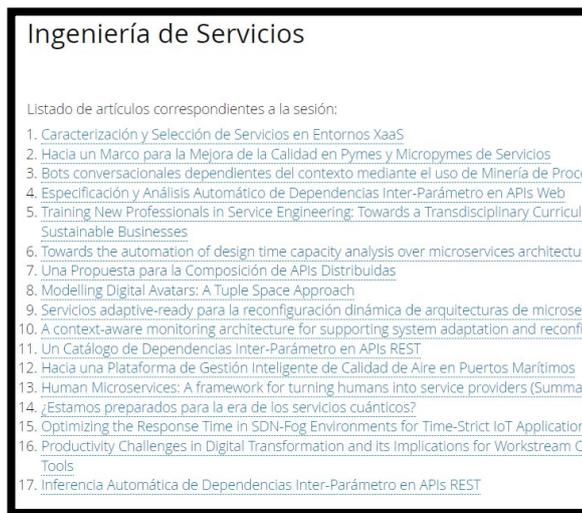


Figura 25: Interfaz de WordPress

En *JCIS 2021* por ejemplo, encontramos estos *tracks*: Preliminares, Comités, Ingeniería de Servicios, Cloud e IoT, Gerontotechnology Forum, Blockchain Forum, Ingeniería de Procesos, Demostraciones de Herramientas. En este caso, Preliminares y Comités se encargan simplemente de introducir los documentos y describir la sesión. Según la estructura de *OJS*, se ha optado por cargar estos dos como archivos *PDF* dentro de cada issue. En el fichero *XML* aparecen como galeradas (*galley*s) codificadas. Para el resto de *tracks*, se sigue la estructura que marca el módulo de importación de *OJS*. Esta estructura se entiende fácilmente y es bastante intuitiva.

Un ejemplo de *track* es Ingeniería de Servicios, en la sesión de 2021 de *JCIS*. *OJS* necesita que cada issue cree sus secciones dentro del elemento `<sections>`.

Ahora, cuando se importe el fichero con todos los artículos de esta sesión, habrá que detallar la referencia de la sección donde queremos que se guarde el mismo. Si por ejemplo hemos asignado la referencia “is” a Ingeniería de Servicios, la manera correcta de asignarlo sería `section_ref="is"`. De esta forma, cuando se sube el archivo, automáticamente se guardan los artículos en los *tracks* correspondientes. La vista que ofrece *OJS* de la interfaz es la siguiente:

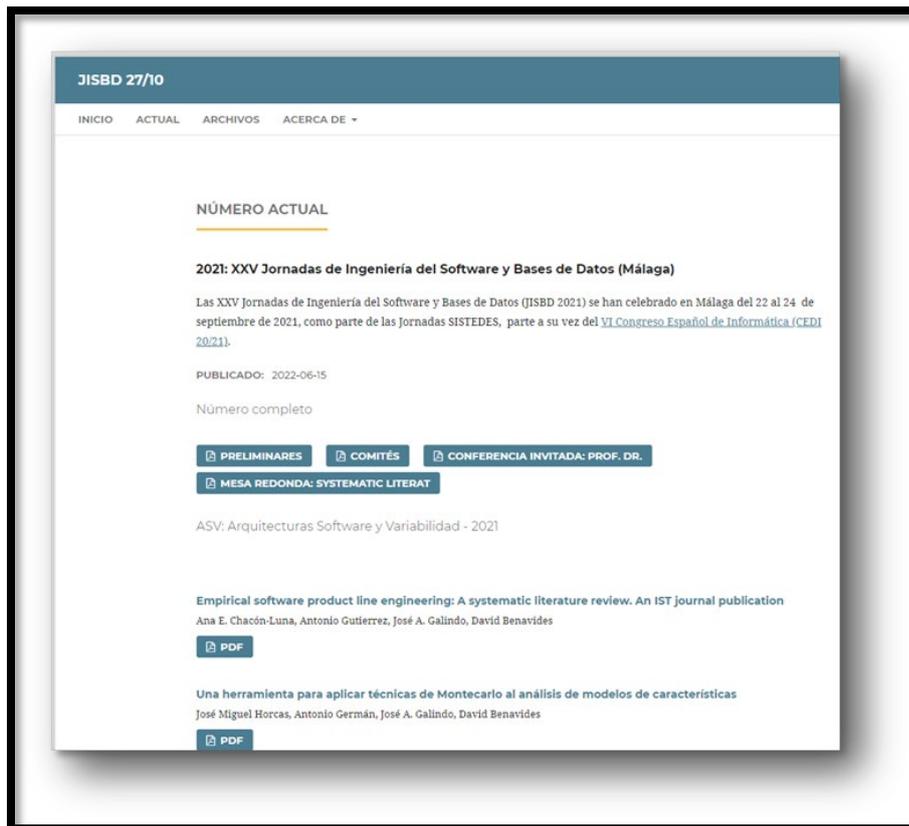


Figura 26: Vista interfaz de JISBD en OJS

*Número completo* corresponde con los *PDF* de información de sesión, como puede ser Preliminares, Comités, y conferencias invitadas. Posteriormente, se muestra cada *track* con los artículos de cada uno. Desde esta pantalla se puede acceder a la información de cada artículo.

```

<sections>
  <section ref="a1" seq="1" editor_restricted="0" meta_indexed="1"
    <id type="internal" advice="ignore">210</id>
    <abbrev locale="es_ES">a1</abbrev>
    <policy locale="es_ES">Política de sección por defecto</policy>
    <title locale="es_ES">Sesión 1: Knowledge discovery / Symbolic
  </section>
  <section ref="a2" seq="2" editor_restricted="0" meta_indexed="1"
    <id type="internal" advice="ignore">211</id>
    <abbrev locale="es_ES">a2</abbrev>
    <policy locale="es_ES">Política de sección por defecto</policy>
    <title locale="es_ES">Sesión 2: Query languages</title>
  </section>
  <section ref="a3" seq="3" editor_restricted="0" meta_indexed="1"
    <id type="internal" advice="ignore">212</id>
    <abbrev locale="es_ES">a3</abbrev>
    <policy locale="es_ES">Política de sección por defecto</policy>
    <title locale="es_ES">Sesión 3: Specification and implementati
  </section>
  <section ref="a4" seq="4" editor_restricted="0" meta_indexed="1"
    <id type="internal" advice="ignore">213</id>
    <abbrev locale="es_ES">a4</abbrev>
    <policy locale="es_ES">Política de sección por defecto</policy>
    <title locale="es_ES">Sesión 4: Quantum computing</title>
  </section>
  <section ref="a5" seq="5" editor_restricted="0" meta_indexed="1"
    <id type="internal" advice="ignore">214</id>
    <abbrev locale="es_ES">a5</abbrev>
    <policy locale="es_ES">Política de sección por defecto</policy>
    <title locale="es_ES">Sesión 5: Testing</title>
  </section>
  <section ref="a6" seq="6" editor_restricted="0" meta_indexed="1"
    <id type="internal" advice="ignore">215</id>
    <abbrev locale="es_ES">a6</abbrev>
    <policy locale="es_ES">Política de sección por defecto</policy>
    <title locale="es_ES">Sesión 6: Fuzzy logic programming / Veri
  </section>

```

Figura 27: Secciones en XML

### 5.2.3 Archivos PDF

Para llevar a cabo esta parte del trabajo, se ha accedido al fichero de configuración de *PHP* con el fin de cargar archivos de tamaño mayor al que viene por defecto. Para ello, se edita el fichero *php.ini*.

Existen 3 variables que interesan cambiar valores para hacer una carga de contenido de este tipo:

*Upload\_max\_filesize*

*Post\_max\_size*

*Max\_file\_uploads*

Por defecto, *PHP* deja subir archivos de como máximo 2MB. En nuestro caso, hemos escogido hacer una migración por lotes, pero con la cantidad de artículos y ficheros a cargar que existen, disponemos de algunos con más de 15MB. Una vez actualizado el fichero ha sido posible seguir con la migración.

Esta parte de la obtención del fichero de importación ha sido la más compleja, ya que se han tenido que resolver dos problemas. El primero de ellos ha sido ver de qué forma poder generar el enlace correspondiente al *PDF* de cada artículo, y en segundo lugar obtener el formato de archivo correcto para la importación. Una de las formas que tiene *OJS* de cargar un documento es mediante la codificación en *base64*.

Para la generación del enlace que contiene el documento que se quiere codificar se tienen dos tipos de direcciones. Dependiendo del año de publicación, los archivos se cargaron en un destino diferente. Para 2015 y 2016 se encuentran en la ruta `‘/wp-content/uploads/<año>/./’`. Esta ubicación no tiene una lógica que relacione el nombre de la ruta con el artículo de *WordPress*. Estos dos años de documentos se han añadido manualmente. A partir de 2017, la ruta de acceso de la mayoría de documentos sigue el patrón `‘/submissions/<paper_PDF_file>’`, donde el valor de la *key* `‘paper_PDF_file’` nos devuelve la ruta completa del *PDF* que se busca.

Una vez terminada la primera parte del proceso, hay que obtener el *PDF* codificado. La forma elegida en este trabajo ha sido a través de un programa en *PHP*. Para trabajar con los documentos *XML* se hace uso de la clase *DOMDocument*. Esta clase representa un documento *XML* en su totalidad, y sirve como raíz del árbol del documento. Tiene varias funciones útiles para trabajar con estos ficheros.

Este programa carga en primer lugar el *XML* generado con todos los enlaces. Gracias a la función `getElementsByTagName()`, se guarda en una variable todos los elementos que coinciden con el nombre, en este caso *embed*. Para cada elemento se guarda el valor del nodo correspondiente y se codifica el contenido obtenido del enlace. *OJS* necesita también el tamaño de archivo codificado, que se guarda gracias a la función `‘strlen()’` del contenido del enlace.

Para insertar un documento en un artículo mediante *XML*, se emplea el elemento `‘<submission_file>’` y `‘<article_galley>’`. El elemento `‘submission_file’` contiene el *PDF* codificado con sus atributos, nombre y tamaño de archivo. El elemento *article\_galley* se emplea para crear una galería de documentos para el artículo. Cabe recalcar que es necesario asignar el identificador correcto del *submission\_file\_ref* para que *OJS* asigne de manera correcta el *PDF* a la galería creada.

#### **5.2.4 Autores**

La carga de autores dentro de *OJS* es posible a través del artículo en los que han participado cada uno. Para esta parte del proyecto se ha empleado otro mapeo.

A la hora de añadir los autores de cada artículo, se comprobó que el mapeado empleado no conseguía añadir correctamente todos los datos solicitados por *OJS*. Esto es debido a que desde *WordPress* se obtienen los metadatos de cada autor en elementos separados, pero compartiendo un identificador, un número en este caso.

```
<wp:meta_key>![CDATA[author_name_1]]</wp:meta_key>
<wp:meta_value>![CDATA[Jose Garcia-Alonso]]</wp:meta_value>
</wp:postmeta>
      <wp:postmeta>
        <wp:meta_key>![CDATA[author_email_1]]</wp:meta_key>
        <wp:meta_value>![CDATA[jgaralo@unex.es]]</wp:meta_value>
      </wp:postmeta>
      <wp:postmeta>
        <wp:meta_key>![CDATA[author_univ_1]]</wp:meta_key>
        <wp:meta_value>![CDATA[University of Extremadura - Spain]]</wp:meta_value>
      </wp:postmeta>
      <wp:postmeta>
        <wp:meta_key>![CDATA[author_name_2]]</wp:meta_key>
        <wp:meta_value>![CDATA[Javier Berrocal]]</wp:meta_value>
      </wp:postmeta>
      <wp:postmeta>
        <wp:meta_key>![CDATA[author_email_2]]</wp:meta_key>
        <wp:meta_value>![CDATA[jbero1m@unex.es]]</wp:meta_value>
      </wp:postmeta>
      <wp:postmeta>
        <wp:meta_key>![CDATA[author_univ_2]]</wp:meta_key>
        <wp:meta_value>![CDATA[University of Extremadura - Spain]]</wp:meta_value>
      </wp:postmeta>
      <wp:postmeta>
        <wp:meta_key>![CDATA[author_name_3]]</wp:meta_key>
        <wp:meta_value>![CDATA[Borja Rivero Jimenez]]</wp:meta_value>
      </wp:postmeta>
      <wp:postmeta>
        <wp:meta_key>![CDATA[author_email_3]]</wp:meta_key>
        <wp:meta_value>![CDATA[brivero@unex.es]]</wp:meta_value>
      </wp:postmeta>
```

Figura 28: Metadatos WordPress

Se puede ver la estructura que sigue en la figura 28. Se identifica a un autor con su nombre, la institución a la que pertenece y su correo electrónico.

La solución propuesta en este trabajo ha sido la inserción aislada de los metadatos de cada autor. En primer lugar, se ha creado un autor con cada email encontrado. Para trabajar con un documento correctamente estructurado, se ha empleado el fichero .csv que se consigue descargando el listado de todos los autores en la base de datos de *WordPress*. Se distinguen tres columnas con los tres identificadores propios.

En este punto, se mapea el documento con los artículos correctamente estructurados para *OJS* y el csv con los metadatos de los autores. Gracias al correo electrónico se puede enlazar los datos de un autor comprobando que coincide su correo.

Se han solucionado muchas inconsistencias de datos en esta parte ya que comprobando que se indexaba correctamente cada dato aparecían problemas con nombres vacíos, caracteres no deseados en los datos, datos mezclados y otros factores que han dificultado la carga de los autores.

Después de terminar con las transformaciones necesarias, cada elemento de origen se asigna al destino, pudiendo crear una tabla con la información resumida de los cambios en los metadatos:

Dato esquema origen	Transformación(es)	Dato esquema destino
title (item)	-	title (publication)
content:encoded (item)	-	abstract (publication)
author_name_* (meta_key)	separar el nombre del apellido, mediante el espacio	givenname (author) familyname (author)
author_univ_* (meta_key)	-	affiliation (author)
author_email_* (meta_key)	-	email (author)
handle (meta_key)	-	id (publication)
keywords (meta_key)	-	keywords (article)
-	PDF codificado en base64	embed (submission_file)
-	-	filesize (submission_file)

Tabla 1: transformaciones de la migración de BDSISTEDES

## 5.3 Carga de datos

El *plugin XML Nativo* ofrece diferentes *plugins* que permiten la inyección de datos y su extracción, así como otros para muchas aplicaciones. En este trabajo nos va a interesar el *plugin* de importación y exportación.

Es posible importar rápidamente mediante el *plugin*. Para utilizar este sistema es necesario un conocimiento básico de *XML* y estar inscrito como Journal Manager en la revista. Hay diferentes archivos ejemplo de muestra para los metadatos y esquemas para su uso en las aplicaciones software.

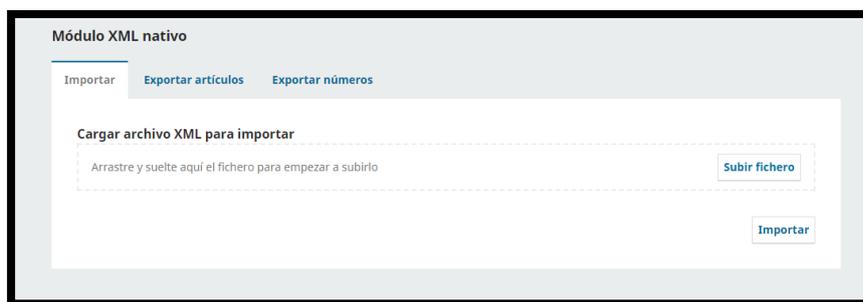


Figura 29: Módulo XML Nativo

Para la migración de los artículos ya existentes, se ha utilizado este *plugin* como módulo de importación. Una vez obtenido el archivo *XML* con el formato adecuado, simplemente se debe subir mediante este *plugin* y los artículos se cargan en la revista. También permite la exportación de artículos y de *issues*, pudiendo transferir a otras revistas o plataformas.

La jerarquía de la carpeta final para esta importación sigue la siguiente lista:

- BOLETINES
  - boletines.XML
- JCIS
  - issuesJCIS.XML
  - JCIS 2015.XML
  - JCIS 2016.XML
  - JCIS 2017.XML
  - JCIS 2018.XML
  - JCIS 2019.XML
  - JCIS 2021.XML
- JISBD
  - issuesJISBD.XML
  - JISBD 2015.XML
  - JISBD 2016.XML
  - JISBD 2017.XML
  - JISBD 2018.XML
  - JISBD 2019.XML
  - JISBD 2021.XML
- PROLE
  - issuesPROLE.XML
  - PROLE 2015.XML
  - PROLE 2016.XML
  - PROLE 2017.XML
  - PROLE 2018.XML
  - PROLE 2019.XML
  - PROLE 2021.XML
- SEMINARIOS
  - seminarios.XML

Antes de la carga, es necesaria la configuración completa del nuevo software. Se deben de crear las revistas oportunas, se deben añadir los *plugins* necesarios y asignar correctamente los roles de cada usuario de la página.

Para ello, en primer lugar se crean las cinco revistas que van a ser usadas en la biblioteca de SISTEDES. Como se ha comentado anteriormente, tendremos para los Seminarios, Boletines, *JISBD*, *JCIS* y *PROLE*.

Ahora, en las tres conferencias se añaden los documentos que proporcionan la información necesaria para crear los “issues” y secciones, dividiendo correctamente cada artículo en la sección y en el año correspondiente.

Antes de la carga de los artículos es necesario activar los *plugins* necesarios. El *plugin* más importante en este caso es el de asignación de identificadores a cada artículo. En este caso, todos los años anteriores hasta 2022 se cargan con el handle ya asignado, como se ha comentado en el apartado de handles. Una vez activado, cuando se cargue el metadato correspondiente al identificador, aparecerá correctamente el handle en el artículo a mostrar.

El *plugin* de los identificadores de artículo ha sido modificado para poder emplear el formato adecuado a la revista de SISTEDES. Ha sido necesario comentar ciertas líneas

de código sobre comprobaciones en los formatos, y nomenclaturas a la hora de llamar a cada handle. También se puede elegir el tema de la interfaz que se desee, entre muchos gratuitos o de pago en internet.

Posteriormente, se carga año a año los artículos de cada conferencia, en su respectiva revista. También los seminarios y los boletines.

Como ejemplo, se muestra en la figura 30 la vista principal de una conferencia de la biblioteca, en su número actual y con la figura 31 un artículo aleatorio.

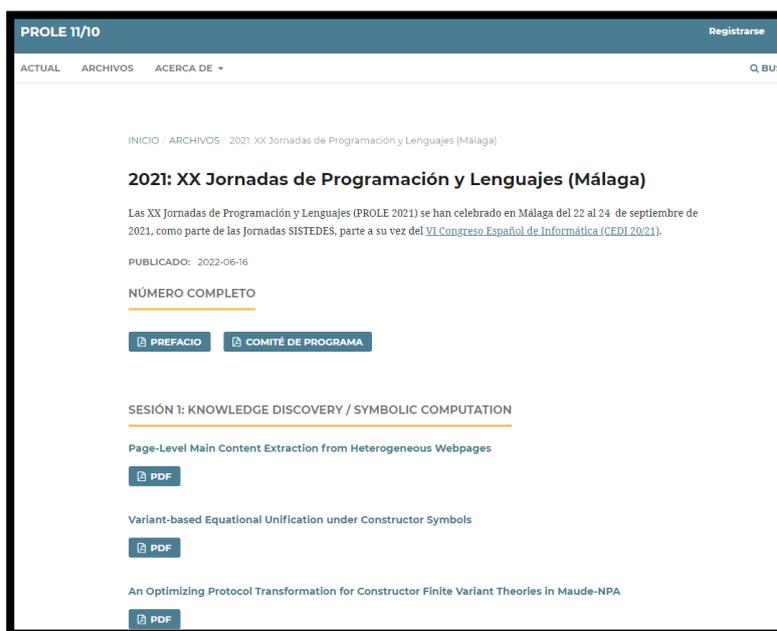


Figura 30: Vista interfaz de PROLE en OJS



Figura 31: Vista interfaz de un artículo en OJS



# CAPÍTULO 6 - Migración 2: Carga anual

---



El proceso que se ha seguido para importar todos los documentos de años anteriores no es posible pensando en el futuro de la biblioteca. La migración de datos se ha hecho desde *WordPress*.

Toda la gestión interna de la biblioteca sobre las conferencias se realiza a través de *EasyChair*. Como ya se mencionó en esta memoria, se trata de un sistema de administración en línea de conferencias académicas. Desde esta plataforma, la obtención de los metadatos es completamente diferente a la usada para hacer la migración anterior.

A continuación, se detalla la estructura de datos a convertir y el proceso empleado para ello.

## 6.1 Descripción de la fuente de datos

*EasyChair* proporciona herramientas para la exportación de sus documentos. Como está especializada en conferencias y documentación de este estilo, es mucho más sencillo sacar partido a estos metadatos.

La fuente de datos separa los datos según la conferencia, por lo que se tienen tres directorios: *JCIS*, *JISBD* y *PROLE*. En cada directorio se encuentran los documentos .csv correspondientes a los metadatos de cada artículo, y otro con los autores participantes en ellos. Cada uno de estos directorios a su vez contiene otro directorio llamado ‘papers’. En este último se encuentran todos artículos en formato PDF. Es importante recalcar que existe un identificador numérico para cada artículo, por lo que resulta bastante sencillo entrelazar de nuevo estos datos. Por ejemplo, si un artículo tiene el código 6, todos sus autores y el *paper* tendrán ese número en su elemento.

El fichero ‘submissions.csv’ contiene información sobre los artículos de una conferencia en el año seleccionado. Desde este archivo se recuperan los siguientes metadatos:

- ❖ Número de *submission*, para enlazar con los otros ficheros.
- ❖ Título del artículo.
- ❖ Las *keywords* o palabras clave.
- ❖ La descripción del artículo.

Las palabras clave aparecen separadas por saltos de línea, por lo que empleando el número de carácter al que pertenece, se ha dividido el *string* en tantas keywords como contenía. El número de artículo se ha utilizado para comprobarse con el documento de autores y con el de los *PDF*. El título y la descripción no tienen diferente formato, por lo que una vez encontrado el número de artículo coincidente, se indexa en el *XML* final.

El siguiente documento es ‘authors.csv’. En este documento encontramos todos los autores con las siguientes columnas:

- ❖ ‘Submission #’, el número de artículo.
- ❖ ‘First name’, nombre.
- ❖ ‘Last name’, apellidos.
- ❖ ‘Email’, correo electrónico.
- ❖ ‘Country’, país.
- ❖ ‘Affiliation’, afiliación/universidad.
- ❖ ‘Web page’, página web.
- ❖ ‘Corresponding’.

Empleando ‘Submission #’, se obtienen todos los metadatos del mismo elemento, es decir, con una simple comprobación se generan todos los autores de un artículo, con todos los datos necesarios.

Es interesante destacar que los nombres tienen el mismo formato que *OJS*, los apellidos separados por un espacio en el mismo campo. Respecto a la anterior migración, resultó complicado el volcado de este tipo.

Para asignar cada *PDF* en el artículo correspondiente se utiliza el directorio ‘papers’. En esta parte de la carga anual también se contemplan cambios significativos en el proceso de transformación de los datos. Dentro del directorio aparecerán todos los documentos en formato *PDF* de los artículos. Un ejemplo del nombre de un documento en esta carpeta sería el siguiente: ‘2022\_JCIS\_7.PDF’.

Primero aparece el año de la conferencia, después las siglas, y por último el número de artículo al que pertenece. Si se recuerda de la migración de los datos legados, los *PDF* codificados en *base64* se obtenían desde la dirección *URL* donde estaban almacenados, en la página de *WordPress*. Desde ahí se descargaba el documento, se codificaba y se añadía al fichero *XML*. Ahora se tiene un directorio con todos los documentos guardados.

Como se está trabajando con archivos *XML* desde el principio, se ha empleado un programa en *PHP* que crea de manera sencilla un documento *XML* con todos los *PDF* que contenga un cierto directorio, todos ellos codificados en *base64*. El programa crea un elemento `<file>` para cada artículo y guarda su nombre y el tamaño del fichero.

Con estos datos, se trabaja de manera sencilla en *MapForce*, sabiendo que se han añadido los ficheros anteriores al trabajo. Así, con simples comprobaciones de números enteros, se genera el documento final para su posterior carga en *OJS*.

## 6.2 Definición de la transformación

Para conseguir el documento final de subida, se ha empleado *MapForce* como mecanismo de transformación y adaptación de los datos. La forma de conseguirlo se detalla a continuación.

Este software trabaja sobre esquemas de *XML*, por lo que si se tiene un esquema destino que se desea obtener, aporta una manera visual de hacerlo.

En primer lugar se han cargado los tres documentos comentados anteriormente, seleccionando de cada uno los elementos interesantes de guardar.

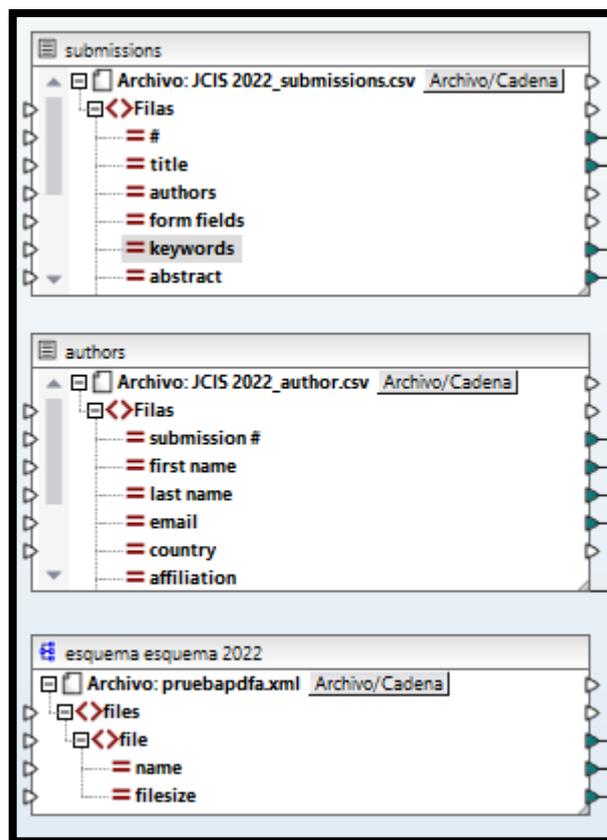


Figura 32: vista MapForce carga anual

En la figura 32 se observa qué datos interesan a la hora de transformar estos datos. En esta imagen, el archivo 'pruebaPDFa.XML' contiene todos los PDF codificados dentro de un mismo documento, gracias al programa PHP creado. Así, con el número de artículo se conectan y permiten generar el documento final para la subida.

En JISBD y PROLE, se añade otro documento correspondiente a los tracks, comentados en el anterior apartado de la migración de datos legados. Aparece de nuevo el identificador de artículo, por lo que se comprueba que coincide y se le asigna el identificador del track que le vincula. Se debe acompañar este paso con la configuración en OJS. Se puede añadir la información de los tracks al XML cargado con la información de los números, o simplemente utilizar la opción de la interfaz que ofrece crear una nueva sección. Es importante que coincida la abreviación elegida en la transformación para vincular correctamente los tracks.

Tras probar la importación, se recalca que hay ciertos valores que tienen que coincidir, ya que OJS emplea referencias de identificación para conectar ciertos argumentos. Por ejemplo, para unir un PDF con su artículo, existe un 'submission\_file\_ref' que actúa como identificador de la publicación, también el propio 'file\_id'. Si no se tiene en cuenta que deben coincidir, en la interfaz de OJS no se verá reflejada correctamente la importación de los datos. Como solución, se utilizan contadores independientes, pero

con mismos valores, incrementándose en una unidad cada vez que se cambia de elemento.

Como en los 'papers' tenemos el nombre con la estructura ya comentada, se separa el string para obtener solamente el número del artículo. Así, con una simple función 'equals' es posible generar el documento correctamente. De este documento guardaremos el tamaño del archivo junto con el propio *PDF* codificado.

Los *handles* se asignan en la transformación de datos, y como en la anterior migración, se cargarán en la plataforma de asignación de *handle*. La metodología es la misma, mediante el identificador de artículo se crea, con la misma estructura, el *handle* del mismo.

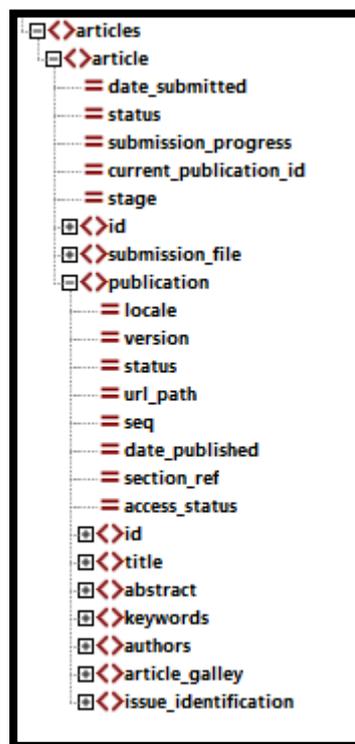


Figura 33: Fichero final carga anual

Al terminar con la conversión, se puede resumir en esta tabla las conversiones y asignaciones de los metadatos:

Dato esquema origen	Transformación(es)	Dato esquema destino
title (submissions.csv)	-	title (publication)
abstract (submissions.csv)	-	abstract (publication)
first name (authors.csv)	-	givenname (author)
last name (authors.csv)	-	familyname (author)
author_univ_* (authors.csv)	-	affiliation (author)
author_email_* (authors.csv)	-	email (author)
-	-	id (publication)
keywords (submissions.csv)	separar cada keyword, utilizando el salto de línea	keywords (article)
file ( <i>papers.xml</i> )	-	embed (submission_file)
name ( <i>papers.xml</i> )	-	name (submission_file)
filesize ( <i>papers.xml</i> )	-	filesize (submission_file)
-	asignación handle con identificador	id (publication)

Tabla 2: transformaciones de la migración de EasyChair

## 6.3 Carga de datos

Una vez creado nuestro archivo final de importación, se comprueba que todo esté correcto con *OJS*. Es posible encontrar algún fallo de estilo. El fallo más común suele ocurrir cuando un artículo no tiene un ‘paper’ asociado, ya que se crea un elemento que no contiene nada, un elemento vacío. *OJS* tiene un sistema de errores no muy avanzado, ya que simplemente informa de que existe un error con el elemento que lo produce, pero no informa en qué línea o en qué elemento concreto ocurre.

En esta carga, el contenido de subida se reduce simplemente a un fichero por conferencia, ya que se carga el año actual, 2022. Este proceso es el esperado para hacer los próximos años. En la anterior migración, resultó más complicado el proceso ya que aparte de ser desde *WordPress*, se trataron datos desde 2015.

Simplemente se cargarán tres ficheros, cada uno de ellos dentro de la revista ya creada anteriormente en la importación de los datos legados:

- ❖ *JCIS 2022.XML*
- ❖ *PROLE 2022.XML*
- ❖ *JISBD 2022.XML*

El proceso se realizaría una vez al año, y se reduciría simplemente a conseguir todos los ‘papers’ codificados gracias al programa *PHP*, y la asignación de los datos para el fichero final, esto en cada una de las conferencias que dispone la biblioteca de SISTEDES.

En este punto ya está toda la carga en la aplicación, se puede comprobar el funcionamiento de la biblioteca y los accesos posibles, junto con las funcionalidades. Predeterminadamente, *OJS* prioriza la navegación por revista, por lo que no existe directamente una conexión entre ellas. Ofrece una interfaz común, donde aparecen recopiladas todas las revistas con toda su información.

La ventaja que tiene este software se encuentra dentro del panel del administrador, los menús de navegación. En principio se tienen dos menús, el del usuario, y el primario de navegación. El primario permite las acciones de registrarse, panel de control, entrar, etc. Por otro lado, desde el primario se puede navegar al número actual, a los archivos de la revista, contacto y otro tipo de información.

También es posible añadir muchas otras funcionalidades como ver el equipo editorial, añadir función de buscar, acerca de la revista, y una *URL* remota o página personalizada.

En nuestro caso, tenemos una navegación simple entre las revistas. Se ha añadido un apartado en el menú de inicio, y un apartado de información sobre aspectos importantes de la revista.

# CAPÍTULO 7 - Conclusiones

---



En este proyecto hemos diseñado y ejecutado la migración y la carga anual de la biblioteca de SISTEDES desde su plataforma actual a la plataforma OJS. Con ello, hemos dotado a la biblioteca digital de un entorno para su gestión mucho más adecuado, lo cual redundará en mejores prácticas de administración.

Al ser una migración de datos, se ha dado importancia al proceso de la misma, desde los primeros pasos, a una visión global de las etapas a seguir. Siempre que se habla de una migración de datos, es imprescindible comentar qué seguridad y qué medidas se van a tomar para reducir al mínimo el riesgo de una pérdida de datos o algún error en la importación de los mismos. En este trabajo, se ha dividido el proceso en otros más pequeños, empleando copias de seguridad en todo momento y escenarios artificiales de carga. Se han generado bases de datos locales en las que se ha trabajado con pruebas y prototipos, y bases de datos en servidores que han aportado la parte online de una importación.

En primer lugar, para conseguir que la migración salga como se planea, es necesario mapear los datos de entrada y entender cómo se estructuran los datos a transformar. Es importante estudiar en detalle la estructura de origen para definir correctamente qué datos se van a utilizar, cómo se van a utilizar y cuándo se van a utilizar. Por lo tanto, los datos de entrada se han mapeado como se esperaba. Ha resultado complicado la primera parte de la extracción de los datos, ya que no se ha seguido una misma forma de carga desde 2015, pero separando correctamente cada año con su estructura ha sido posible.

Por otro lado, se ha estudiado el sistema de destino, viendo sus características y las funciones que tiene, a la vez de las limitaciones y problemas que pueden surgir. Al ver los problemas que ofrecía tener un sistema de gestión de revistas en un software no tan específico, se ha podido observar que seleccionar el entorno de trabajo en un proyecto es igual o más de importante que la propia gestión del mismo.

Pese a haber completado la migración exitosamente, quedan mejoras por implementar. Pensando en el futuro de la gestión de esta biblioteca, sería interesante avanzar en este proceso de carga, automatizando al máximo cualquier acción. Es posible generar un programa que conecte ambas plataformas y hacer la carga, o generar una aplicación que gestione de manera personalizada esta carga, eligiendo qué datos interesa migrar y cuáles no.

La relación de este trabajo con los estudios del Grado en Ingeniería Informática es muy alta, cubriendo aspectos de bases de datos, junto con lenguajes de programación, y de ingeniería del software. Al estudiar un software libre como es Open Journal System, es posible entender y aplicar lo estudiado durante la carrera en este proyecto.



# Referencias

- Altova. (2022). *Altova*. Recuperado el 10 de octubre de 2022, de <https://www.altova.com/es/>
- Board, I. G. (2010). *IFLA/UNESCO Manifesto for Digital Libraries*. Recuperado el 10 de noviembre de 2022, de <https://cdn.ifla.org/wp-content/uploads/files/assets/digital-libraries/documents/ifla-unesco-digital-libraries-manifesto.pdf>
- EasyChair. (2022). *EasyChair*. Recuperado el 15 de septiembre de 2022
- OJS. (2022). *Open Journal Systems*. Recuperado el 7 de octubre de 2022, de <https://pkp.sfu.ca/ojs/>
- SISTEDES. (2022). *SISTEDES*. Recuperado el 7 de septiembre de 2022, de <https://www.sistedes.es/>
- Studio, V. (2022). *Visual Studio Code*. Recuperado el 11 de octubre de 2022, de <https://code.visualstudio.com/docs>
- TecnoPymes. (2018). *¿Qué es un CMS – WordPress?* Recuperado el 5 de noviembre de 2022, de <https://www.tecnopymes.org/que-es-un-cms-wordpress/>
- UMA. (2022). *Revistas Científicas*. Obtenido de <https://revistas.uma.es/>
- UPV. (2022). *POLI[PAPERS]*. Recuperado el 10 de octubre de 2022, de <https://polipapers.upv.es/>
- UV. (2022). *Open Journal Systems*. Recuperado el 10 de octubre de 2022, de <https://ojs.uv.es/>
- Wikipedia. (2022). *Handle*. Recuperado el 12 de septiembre de 2022, de <https://es.wikipedia.org/wiki/Handle#:~:text=En%20programaci%C3%B3n%20de%20computadoras%2C%20un,datos%20o%20un%20sistema%20operativo.>
- Wikipedia. (2022). *WordPress*. Recuperado el 5 de octubre de 2022, de <https://es.wikipedia.org/wiki/WordPress>





# Anexo I: Mapping carga anual

A continuación se detalla, como ejemplo, el mapeado de la carga anual en *JCIS*, junto con la estructura de cada documento a mapear.

El documento que contiene todos los *PDF* codificados se ha generado gracias a un programa en *PHP* que tiene la siguiente estructura:

- ❖ Primero se crea un documento *XML* vacío y se añade el elemento *files*.
- ❖ Se recorre toda la carpeta donde se encuentre el programa con un bucle.
- ❖ Se obtiene el contenido de cada archivo '2022\_JCIS\_\$.pdf' donde *\$i* es el número de iteración del bucle.
- ❖ Se crea un elemento *file* con el tamaño, el nombre y el pdf codificado en base64.
- ❖ Se guarda como hijo de *files* con *appendChild()*.
- ❖ Se guarda el documento *XML* una vez terminado el bucle.

Para conseguir los metadatos de los papers se ha empleado la siguiente transformación:

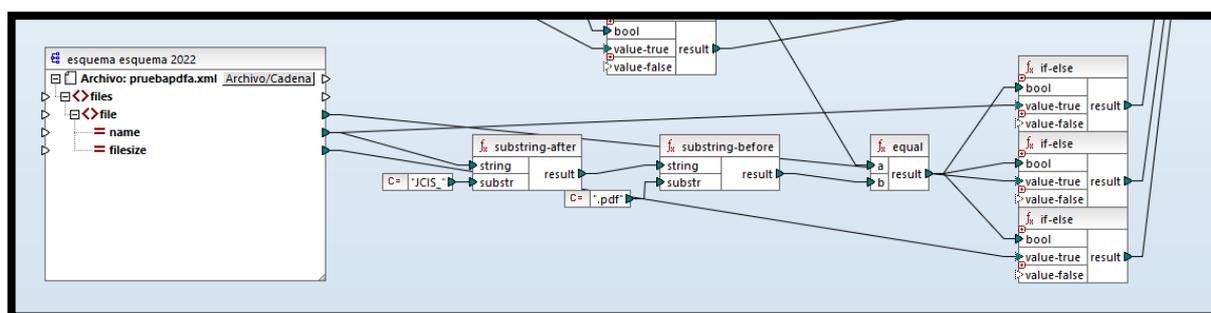


Figura 34: metadatos de los papers (Anexo I)

En *JCIS*, por ejemplo, se comienza seleccionando la parte del *string* después del 'JCIS\_' y antes del '.pdf'. Este substring corresponde al número de *submission* de ese *paper*, que se comprueba con el artículo del documento *submissions*. Al devolver *true*, se escoge el valor de cada elemento que interesa y se añade al documento de importación de OJS.

Sucede algo parecido en el caso de importar los autores y sus metadatos. A través del número de artículo se comprueba que coincida con el correspondiente al autor y se añade al documento final.

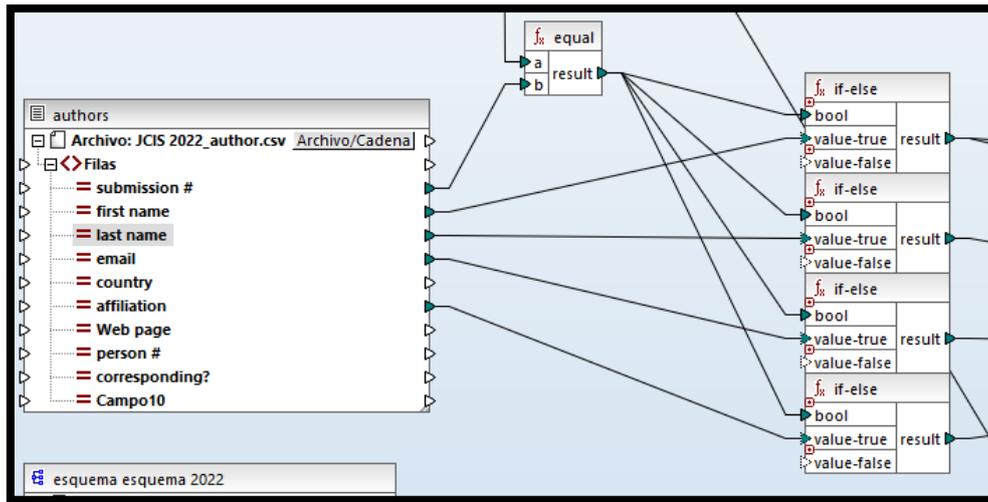


Figura 35: metadatos de los autores (Anexo I)

El contenido de *submissions* está bien tipificado, a excepción de dos elementos. En el caso de las *keywords*, necesitan separarse cada una, esto se ha hecho gracias a la siguiente función:

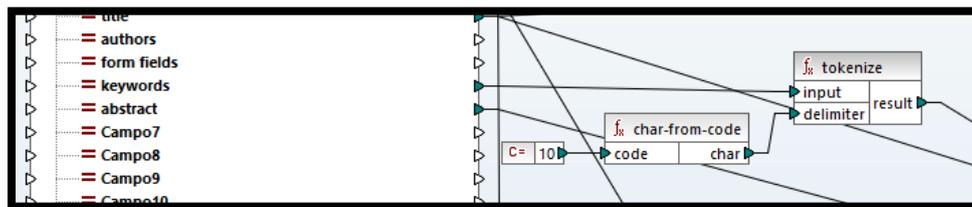


Figura 36: keywords de las submissions (Anexo I)

Se separa cada una con la separación del salto de línea. También, para obtener el handle del documento, se utiliza el número de artículo. Dependiendo de la conferencia y del año, se escoge un string principal, y después se añade el identificador.

*'11705/JCIS/2022/' + id*





# Anexo II: Objetivos de desarrollo sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

<b>Objetivos de Desarrollo Sostenible</b>	<b>Alto</b>	<b>Medio</b>	<b>Bajo</b>	<b>No procede</b>
ODS 1: Fin de la pobreza				X
ODS 2: Hambre cero				X
ODS 3: Salud y bienestar				X
ODS 4: Educación de calidad			X	
ODS 5: Igualdad de género				X
ODS 6: Agua limpia y saneamiento				X
ODS 7: Energía asequible y no contaminante				X
ODS 8: Trabajo decente y crecimiento económico				X
ODS 9: Industria, innovación e infraestructuras		X		
ODS 10: Reducción de las desigualdades				X
ODS 11: Ciudades y comunidades sostenibles				X
ODS 12: Producción y consumo responsables				X
ODS 13: Acción por el clima				X
ODS 14: Vida submarina				X
ODS 15: Vida de ecosistemas terrestres				X
ODS 16: Paz, justicia e instituciones sólidas				X
ODS 17: Alianzas para lograr los objetivos				X

*Tabla 3: Objetivos de Desarrollo Sostenible*

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Los Objetivos de Desarrollo Sostenibles tratan 17 aspectos con el objetivo de mejorar la vida de las personas, buscando un futuro sostenible para todos. Se centran en temas de importancia mundial, desde el fin de la pobreza a temas medioambientales. Este apartado da visibilidad a los problemas actuales de la sociedad, embarcando aspectos de derechos humanos, necesarios para tener una vida digna.

Este trabajo no implica una gran aportación en el conjunto de los objetivos, ya que se trata de un proceso de migración de una biblioteca digital. Sin embargo, se puede relacionar el trabajo con alguno de ellos. Como se observa en la tabla de objetivos, se han relacionado dos objetivos de desarrollo, Educación de calidad e Industria, innovación e infraestructuras. Estos aspectos están centrados en la educación e innovación.

El ODS 4 (Educación de calidad) comenta diferentes metas, siendo una de ellas el aumento de competencias técnicas y profesionales para el acceso al empleo. Se centra en dar importancia a la educación, sobre todo infantil, y en países en desarrollo. SISTEDES aporta en el desarrollo científico y tecnológico, promoviendo la investigación e innovación, por lo que de cierta manera apoya este objetivo. Si se relaciona este apartado con promover la educación y el desarrollo en aspectos de enseñanza, el diseño y la implementación de la migración puede aportar en el objetivo.

El ODS 9 (Industria, innovación e infraestructuras) tiene una meta centrada en modernizar la infraestructura utilizando los recursos con mayor eficacia, por lo que se puede relacionar con la mejora en la gestión de la biblioteca. Al conseguir ventajas migrando el software gracias a este trabajo, se marca un grado medio de influencia en este objetivo.

Como conclusión en este apartado final del trabajo, se ha guardado un lugar a la concienciación de construir un mundo mejor. Pese a la poca relación, cualquier aportación se tiene en cuenta, y si todos ponemos algo de nuestra parte, nos acercaremos más al objetivo final.