

IMPLEMENTACIÓN DE ESTRATEGIAS DE SERVICIOS DE MARCACIÓN Y PUBLICACIÓN DE ARTÍCULOS CIENTÍFICOS EN OJS

Santiago Soler

Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
santiago.soler@sedici.unlp.edu.ar
 <https://orcid.org/0009-0003-0594-9581>

Tomás Nahuel Messineo

Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
tomasmessineo@sedici.unlp.edu.ar
 <https://orcid.org/0009-0007-9086-3609>

Dolores García

Comisión de Investigaciones Científicas;
CESGI, Argentina
dolores.garcia@sedici.unlp.edu.ar
 <https://orcid.org/0000-0002-6686-3138>

Gonzalo Luján Villarreal

Universidad Nacional de La Plata;
PREBI-SEDICI; Comisión de Investigaciones
Científicas; CESGI, Argentina
gonzalo@prebi.unlp.edu.ar
 <https://orcid.org/0000-0002-3602-8211>

Adela Ruiz

Universidad Nacional de La Plata; Coordinación de
Revistas Científicas, Argentina
adelarui@perio.unlp.edu.ar
 <https://orcid.org/0000-0002-2873-006X>

DOI: 10.22477/xiv.biredial.393

EJE TEMÁICO: Comunicación académica, científica y
cultural en abierto

RESUMEN

Con el objetivo de agilizar, automatizar e independizar la marcación de artículos científicos en el contexto de la Universidad Nacional de La Plata, se encararon una serie de estrategias para solucionar los diversos desafíos que plantea la marcación. Para el problema de la marcación de referencias bibliográficas en el plugin docxConverted, se incorporaron funcionalidades clave que permiten una mejor integración de este proceso en el flujo editorial. También se agregaron mejoras en la generación de archivos PDF, lo que permite que estos se adapten a los estándares de diseño y presentación de las revistas de la UNLP. Además, se trabajó en la posibilidad de vincular las citas dentro del texto con las referencias bibliográficas al final del documento.

Palabras-clave: Marcación, interoperabilidad, publicaciones científicas, XML JATS.

ABSTRACT

With the goal of streamlining, automating, and making the markup of scientific articles independent within the context of the National University of La Plata, a series of strategies were undertaken to solve the various challenges posed by markup. To address the issue of bibliographic reference markup in the docxConverted plugin, key functionalities were incorporated to enable better integration of this process into the editorial workflow. Improvements were also made to the generation of PDF files, allowing them to adhere to the design and presentation



standards of the UNLP journals. Additionally, work was done to enable the linking of in-text citations with the bibliographic references at the end of the document.

Keywords: Markup, interoperability, scientific publications, XML JATS.

INTRODUCCIÓN

Actualmente, la generación de XML JATS (Journal Article Tag Suite), estándar técnico que describe la estructura y el contenido de artículos de revistas científicas (Redalyc, 2016a), para los artículos producidos en muchas instituciones académicas y científicas se encuentra fuertemente acoplada a herramientas externas, lo que genera una dependencia con la disponibilidad y capacidades de dichas herramientas así como también con los organismos que las ofrecen. Entre estas herramientas de marcación de XML JATS provistas por terceros se destacan los casos de SciELO-XML, provista por Scielo, y de Marcaly, provista en sus diferentes versiones a través de Redalyc y AmeliCA (Becerril-García *et al.*, 2023; Becerril-García y Aguado-López, 2019).

En este trabajo se introduce un conjunto de herramientas para realizar los procesos de marcación y generación de XML JATS para artículos científicos, que pueden ser gestionadas por las propias instituciones que gestionan dichos artículos, y que integran este proceso de marcación dentro del flujo de trabajo de las revistas académicas mediante el sistema de gestión OJS (Open Journal Systems).

En la actualidad, la dependencia de servicios de terceros para el marcado de artículos y la incompatibilidad entre diferentes sistemas de marcado es un gran desafío con el que deben lidiar muchos editores de revistas científicas, quienes a menudo se ven limitados por las reglas (y los cambios de reglas) impuestas por organizaciones externas. El desarrollo que se presenta en esta ponencia propone abordar estos desafíos mediante la implementación de herramientas de marcado y generación de XML JATS personalizadas, que puedan ser gestionadas por las propias instituciones editoras de revistas científicas, fomentando así la independencia en la gestión editorial y apoyando la adecuación de los procesos editoriales a las necesidades y requerimientos particulares de cada revista.

Este trabajo surge del relevamiento en el que se evalúan diferentes herramientas disponibles en el ecosistema del software OJS para realizar la tarea de marcado (Soler *et al.*, 2024). En dicho relevamiento, se realiza un análisis del proceso de marcado en sí mismo y se propone un proceso compuesto por etapas consecutivas. A partir de esta propuesta, se realizó una evaluación de las herramientas disponibles en OJS para realizar la tarea de cada etapa, observando las limitaciones de cada caso y la factibilidad de realizar cambios en cada una en caso de ser necesario. En dicho trabajo se observó que, si bien estas herramientas ofrecían soluciones para la conversión, la edición y la presentación de documentos científicos, presentaban algu-



nos obstáculos que impedían su puesta en producción en un entorno real. El trabajo aquí presentado detalla las soluciones propuestas para algunos de estos obstáculos y propone líneas de trabajo a futuro para continuar avanzando con las mejoras sobre estas herramientas.

ESTADO DEL ARTE Y HERRAMIENTAS EXISTENTES

El estándar XML JATS (ANSI/NISO Z39.96-2019) es ampliamente utilizado para describir artículos científicos publicados en línea. Permite representar tanto el contenido como los metadatos de los artículos, facilitando su procesamiento automático, la generación en múltiples formatos (PDF, HTML, ePUB), y su preservación digital. En los circuitos editoriales, los artículos aceptados pasan por un proceso de marcado en XML JATS antes de su publicación. Este proceso requiere una plataforma compleja con servicios integrados, como la detección de secciones del texto, el procesamiento de citas, y la vinculación con objetos externos. Debido a esta complejidad, muchas instituciones tercerizan la tarea de marcado utilizando herramientas como Marcalyc (Redalyc/AmeliCA) o el software de Scielo. Estas instituciones ponen a disposición de la comunidad científica su infraestructura para desarrollar, distribuir y/o acceder a estos sistemas, y para ello pueden introducir condiciones para acceder a sus plataformas y utilizar sus herramientas. En ocasiones, estas condiciones pueden presentar limitaciones a la autonomía editorial. Asimismo, se han detectado problemas de compatibilidad entre versiones de XML JATS utilizadas por los distintos sistemas, lo que genera también inconvenientes para aquellas revistas que deseen integrar los artículos marcados en diferentes directorios.

A fin de avanzar hacia una alternativa a estos sistemas, se inició un proyecto orientado al desarrollo y mejora de las herramientas existentes en el software OJS, buscando adaptarlas a las necesidades específicas del flujo editorial de cada revista y de cada equipo editorial. A partir del análisis realizado por Soler *et al.* (2024), se detectaron algunas herramientas, y se identificaron limitaciones que si bien permitían pensar un flujo de marcación de XML JATS, presentaban barreras que dificultan su implementación real. Estas barreras podrían requerir la adecuación de los procesos editoriales establecidos, lo que a su vez podría resultar un obstáculo para su adopción por parte de muchas revistas. Frente a esto, el presente trabajo busca abordar las principales limitaciones, con el fin de facilitar una implementación efectiva. En dicho trabajo se proponen tres ejes o etapas en las que esas herramientas intervienen para identificar en qué etapa del flujo editorial se utilizan: 1) conversión de docx a XML JATS; 2) edición de XML JATS; 3) presentación de documentos. Para cada etapa se destacan las herramientas evaluadas, que pueden verse en la Tabla 1.

**Tabla 1** - Herramientas destacadas para la marcación en OJS

	Eje 1: Conversión a XML JATS	Eje 2: Edición de XML JATS	Eje 3: Presentación
Herramienta/Plugin	docxConverter	Texture Plugin	JATSParserPlugin

A partir de esta clasificación, se analizaron en detalle las limitaciones específicas de cada herramienta. Se comenzó por docxConverter, un plugin de OJS que permite generar un XML JATS a partir de un .docx e integra la generación del XML JATS dentro del flujo editorial. Luego de una etapa de pruebas de conversión y análisis de resultados, se pudieron determinar algunas limitaciones, entre ellas, el marcado y la generación de XML JATS de las referencias bibliográficas. Así, uno de los problemas a tratar en este trabajo será el desarrollo de una herramienta que permita generar el XML JATS de las referencias bibliográficas que se integre a docxConverter.

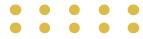
A continuación se analizó JATSParserPlugin, plugin usado en la etapa de generación de los documentos en formato PDF. Entre las principales limitaciones observadas se encontraron la estructura básica del PDF generado y la carencia de opciones de personalización de dicho PDF. Además, como se quiere generar un PDF a partir del XML, se debe asegurar que el PDF resultante se adecúe a las normas de la revista, como ser la incorporación de metadatos en diversos idiomas o la generación de referencias bibliográficas bajo determinado estándar. El detalle de las limitaciones puede verse en la Tabla 2.

Tabla 2 - Limitaciones de las herramientas de marcado

docxConverter	JATSParserPlugin
Generación de XML JATS de las referencias bibliográficas	Plantillas personalizadas
Marcación de notas	Estructura muy básica de PDF
Marcación de fórmulas matemáticas	Vinculación de citas y referencias para humanos
	Generación de referencias bibliográficas según el estándar adoptado por la revista

INTEGRACIÓN DE ESTRATEGIAS PARA LA MARCACIÓN DE REFERENCIA BIBLIOGRÁFICA

En el marco del desarrollo y mejora de las herramientas existentes, con el objetivo de adaptarlas a las necesidades específicas del flujo editorial de las revistas científicas, se propusieron tres desarrollos complementarios. El primero de ellos se enfoca en una problemática



muy común: la necesidad de convertir referencias bibliográficas contenidas en archivos .docx al formato estructurado XML JATS. Más allá de ofrecer una solución puntual para una norma de citación como APA 7ma edición, el eje de este desarrollo fue diseñar una arquitectura modular que permita procesar distintos tipos de referencias de forma extensible, adaptable y compatible con el entorno de trabajo en OJS. Esta estrategia se valida inicialmente con el caso de APA 7ma edición, pero está pensada como una solución escalable, capaz de incorporar otras normas de citación y enriquecer automáticamente los metadatos mediante servicios como OpenAlex.

docxConverter es un plugin para OJS 3.3 que permite integrar una herramienta de conversión de .docx a XML JATS, denominada docxToJats, que se encarga de convertir y generar el XML a partir de un .docx. Las limitaciones en la conversión de documentos descritas en este trabajo hacen referencia a limitaciones de esta herramienta puntualmente. docxToJats analiza los documentos en formato docx e identifica, dentro del contenido de cada documento, los elementos que deberá incorporar en cada una de las secciones de un documento en formato XML JATS, a saber:

1. front -> metadatos de la revista/artículo
2. body -> contenido del artículo
3. back -> referencias bibliográficas

La sección *front* contiene los metadatos del artículo, que pueden ser obtenidos del propio OJS y por lo tanto no se consideran desde el documento original en formato docx. Esto implica que, al momento de realizar la marcación, se trabaja puntualmente sobre las secciones *body* y *back* del XML JATS. La diferencia entre esas secciones es que, en general, el *body* tiene elementos relacionados con el contenido del artículo; en cambio, en el *back* se encuentra la información de las referencias bibliográficas, y suele ser una de las secciones que lleva más tiempo marcar (Redalyc, 2016b; National Information Standards Organization, 2019; McGlone, 2013; Bösch, 2021). Uno de los objetivos propuestos en este proyecto fue la automatización del proceso de marcado de referencias bibliográficas, a fin de disminuir significativamente el tiempo total de generación de XML JATS. Asimismo, otro objetivo planteado fue la automatización del procesamiento de la sección *body* a partir del documento original, considerando tanto secciones como tablas e imágenes.

En resumen, el objetivo fue desarrollar una herramienta que pueda generar el *back* de un XML JATS a partir de las referencias bibliográficas dadas en un .docx y, una vez creada la herramienta, incorporarla a docxToJats para obtener así un XML JATS que integre *front*, *body* y *back*.

La generación de la sección *body* requirió el análisis del código fuente de docxToJats, ya que esta herramienta era capaz de automatizar esta etapa siempre y cuando el documento



original cumpla con determinadas características estructurales (uso de encabezados, nombres de sección, etc.). Por lo tanto, se evaluó el proceso de análisis que realiza esta herramienta, y se documentó la estructura esperable para documentos docx que aseguraban su procesamiento automático. Esta información servirá para generar plantillas que los equipos editoriales podrán compartir con sus potenciales autores como parte de las normas de envío de propuestas.

Para la generación del back, la principal dificultad que se encontró fue la diversidad de estándares de citas (APA, AMA, IEEE, etc.) y en determinados estándares la variabilidad en formas citación (autor y año, año, año y páginas). La solución propuesta debía ser lo suficientemente flexible para procesar cualquier estándar de citación; en una primera etapa, se decidió implementar el soporte para APA, en su 7^a versión, ya que es el más utilizado en la institución donde se lleva adelante este proyecto, y esto generará un impacto positivo en un gran número de revistas científicas propias.

En este sentido, será preciso plantear un flujo de trabajo en OJS que integre las herramientas de conversión, edición y generación de formatos a partir de XML JATS.

DESARROLLO DE LA HERRAMIENTA CITATION-PARSER

La idea principal de este desarrollo, que se denominó citation-parser, surge como una iniciativa propia con el objetivo de ampliar las capacidades de la herramienta de generación de XML JATS, docxToJats. Este nuevo componente fue desarrollado específicamente para incorporar la conversión automatizada de referencias bibliográficas, permitiendo así generar un XML JATS más completo, enriquecido con información estructurada y vinculada según los estándares requeridos.

La herramienta desarrollada permite:

1. Reconocer un documento .docx con una marca semántica o una estructura en la que estén definidas las referencias.
2. Identificar las citas que cumplen la norma APA 7 y descomponer cada una en sus partes: autor, fecha, título, URL, entre otras.
3. Disponer de una estructura modular que permite la adición de cualquier otro estilo de citación.
4. Construir un elemento (nodo) XML de cada referencia bibliográfica para incorporarlo a un XML JATS completo.
5. Integrar la herramienta con OpenAlex para reconocer las referencias que posean DOI y así obtener tanto metadatos normalizados como también información adicional de cada referencia.



6. Reconocer errores en las referencias, e indicar oportunamente en caso de que estén mal confeccionadas o no esté soportado el tipo de referencia, sin que esto interrumpe el procesamiento del resto de referencias.
7. Adaptar la salida XML JATS para asegurar su funcionamiento en el plugin Texture para edición de XML.

ARQUITECTURA DE CITATION-PARSER

A continuación, se detalla la arquitectura del desarrollo y las principales decisiones tomadas, en su mayoría con miras a la integración con docxConverter por un lado, y por otro para permitir la extensión del proyecto para que incorporar otros estándares de citación.

Esta herramienta recibe como entrada un .docx con una sección de referencias bibliográficas (“References”), en la que deberán encontrarse todas las referencias bibliográficas bajo el estándar de citación adoptado por la revista (APA 7 en esta versión inicial). A partir de estas referencias, se generará una primera parte del XML JATS, específicamente la etiqueta <ref-list>, que contiene los metadatos de cada referencia bibliográfica; esta porción de XML será posteriormente integrada en el XML JATS completo. Para realizar esta tarea, se dividió el problema en dos partes: 1) la interpretación (parseo) de las referencias bibliográficas, identificando cada una de las partes componentes, y 2) la generación del nodo estándar XML JATS a partir de las componentes identificadas en 1).

En este punto se encontraron dos grandes dificultades: por un lado, el desarrollo permitía hasta este momento procesar referencias en un único estándar, y no estaba diseñado para incorporar otros estándares. Si bien una solución rápida podría haber sido cambiar un estándar por otro, en este proyecto se buscó una solución más general, que permita tener varios sistemas de referencias bibliográficas a la vez, y que cada revista decida cuál desea utilizar para sus artículos. Cabe aquí destacar que el procesamiento de cada cita bibliográfica, independientemente del estándar utilizado, requiere primero identificar el tipo de recurso citado (artículo, tesis, libro, etc.), posteriormente segmentar la cita en sus partes componentes (autores, título, revista o libro, año, etc.), y finalmente generar un nodo XML JATS que incluya los metadatos adecuados según el tipo de recurso citado, como ser el director si se trata de una tesis, o el ISSN si se trata de un artículo de revista.

Para abordar estos desafíos, se desarrolló una arquitectura propia dentro de citation-parser basada en dos componentes clave: los Parsers y los Printers. Los Parsers son responsables de interpretar las referencias bibliográficas, identificar el tipo de fuente a la que se hace referencia (libros, artículos, capítulos, etc.) y extraer sus partes constitutivas. Luego, los Printers utilizan esa información estructurada para generar el bloque correspondiente en formato XML JATS, adaptándolo al tipo de elemento detectado. Esta división modular fue diseñada



específicamente por nuestro equipo para facilitar la extensión futura del sistema, permitiendo incorporar nuevos tipos de citas o estilos sin necesidad de reescribir la lógica completa.

Así, por un lado se obtendrán múltiples Parser capaces de reconocer referencias bibliográficas en APA 7ma edición, y en un futuro se podrán definir distintos Parsers capaces de reconocer otras estándares de citación (IEEE, AMA, etc), y por otro lado cada Parser es capaz de reconocer cada fuente referenciable en APA 7ma edición, esto quiere decir que puede reconocer la estructura de una referencia bibliográfica de un libro, revista, congresos, etc. Y en un futuro se podrán agregar o modificar.

LIMITACIONES DE CITATION-PARSER

Como se mencionó, existe una complejidad inherente a la hora de procesar referencias bibliográficas, tanto debido a la existencia de múltiples estándares (APA, IEEE, AMA, etc.) como a la estructura definida para cada referencia bibliográfica de acuerdo a su tipología documental (artículo, tesis, libro, etc), independientemente del estándar de citación. En el contexto de este trabajo, se optó por priorizar los elementos autor, fecha, título y URL de las referencias bibliográficas. Cabe aquí destacar que estos elementos tienen a su vez diferentes interpretaciones; por ejemplo, el autor puede ser tanto una persona física como una institución, o el título puede contener el título del artículo, el nombre de la revista, volumen y número si se trata de una revista, o título del libro si se trata de un libro. Sumado a esto, pueden surgir problemas al procesar citas incompletas; un ejemplo de esto es el uso de la inicial en los nombres de autores (en lugar del nombre completo) en APA 7, con lo cual al momento de generar un XML JATS se pierde información potencialmente relevante. La solución propuesta a este problema se basa en la integración de citation-parser con el sistema OpenAlex, lo cual se describe en la siguiente sección. Finalmente, existe una limitación relacionada con las citas en cuerpo del documento, ya que el conversor de docxConverter no es capaz de enlazar la cita con la referencia a partir del texto; de momento, esta limitación se resuelve vinculando las citas manualmente con el plugin Texture.

INTEGRACIÓN CON OPENALEX

OpenAlex es una base de datos bibliográfica de acceso abierto que cataloga artículos científicos, autores, instituciones, revistas y conceptos. Lanzada en 2022 por OurResearch (Priem *et al.*, 2022) se presenta como una alternativa libre a plataformas comerciales como Web of Science y Scopus. Su importancia radica en que permite el acceso gratuito y transparente a la información académica, facilitando la investigación y el análisis bibliométrico. Además, ofrece una API pública que permite obtener datos estructurados sobre publicaciones, citas y métricas de impacto, lo que la convierte en una herramienta clave para desarrolladores, biblio-



tecólogos e investigadores en el ámbito de la ciencia abierta. La integración del citation-parser con OpenAlex permite evitar la pérdida de información relevante en determinados estándares de citas. El siguiente ejemplo muestra una referencia correcta en APA 7:

Becerril-García, A., Aguado López, E. y Macedo García, A. (2023). MarcalyC: software para la marcación XML JATS para las revistas científicas de acceso abierto diamante. *Palabra Clave*, 12(2), e179. <https://doi.org/10.24215/18539912e179>

Como se puede observar, en el ejemplo anterior se desconoce el nombre completo del autor o de la institución donde se realizó la producción. Esta información puede obtenerse desde la API de OpenAlex a partir del DOI del recurso, y con ello construir un XML JATS más completo. Para ello, se incorporó un módulo integrado al citation-parser que, luego de reconocer las componentes de cada referencia bibliográfica, identifica el DOI y se conecta a OpenAlex para obtener metadatos sobre la referencia, que luego son utilizados para mejorar los metadatos preexistentes y posteriormente generar un XML JATS más completo. Existe evidentemente una limitación con esta solución, dado que solo se puede aplicar con los elementos referenciales que tienen DOI. En el futuro, será posible incorporar otros sistemas de identificación soportados por OpenAlex, como ser PMID, PMCID y MAG (Velez-Estevez *et al.*, 2023), y quizás algún día sea posible trabajar con identificadores descentralizados, como por ejemplo ARK o DARK (Segundo *et al.*, 2023).

ADAPTACIÓN DE SALIDA DEL XML PARA TEXTURE

Una vez generado el XML JATS a partir del conversor docxToJats, se deberán realizar ajustes sobre el documento. Las causas posibles son muy diversas: el conversor de XML puede haber introducido errores, o el documento original en formato docx no estaba correctamente estructurado, o es necesario incorporar información ausente en el documento original, entre otras. Para resolver esto, existe dentro de OJS una solución robusta y fácilmente integrable al flujo editorial, conocida como Texture Plugin, que ofrece una interfaz amigable para la edición de documentos de textos, abstrayendo al usuario de la estructura XML subyacente. En el trabajo aquí presentado no se introdujeron cambios en esta herramienta; sin embargo, fue necesario realizar modificaciones en el XML generado por docxToJats a fin de asegurar que pueda editarse correctamente mediante Texture. Esto se debe a que el plugin de Texture requiere un XML con una estructura rígida, especialmente en lo relativo a las referencias bibliográficas, y en caso de no respetar esta estructura el documento XML no se visualiza en Texture (se muestra una pantalla en blanco).



GENERACIÓN DE PDF A PARTIR DE XML JATS EN JATS PARSE: UNA ESTRATEGIA PARA PLANTILLAS PERSONALIZADAS

En la última etapa del flujo editorial de OJS —etapa de producción—, una vez que se cuenta con un XML JATS completo, generado a partir de un documento .docx y corregido con herramientas como Texture, se buscará exportar el artículo en diferentes formatos, como ser PDF, HTML, ePub, entre otros. Esta variedad de formatos es esencial para favorecer la visibilidad y accesibilidad de los artículos en distintos dispositivos (computadoras, celulares, tablets, lectores digitales). La generación de PDF, por lo general, es realizada por cada revista científica ya sea mediante la edición manual con software de edición digital y plantillas personalizadas, o automatizando este proceso a partir de los XML JATS con sistemas como Marcalyc. La edición manual permite plantillas que se adecúen a los requerimientos puntuales de cada revista, pero demanda mucho trabajo por parte de los maquetadores de artículos; la generación automática es mucho más fácil, pero se ve limitada por las opciones de personalización de la plantilla disponibles por los sistemas utilizados.

Para abordar esta problemática de manera estratégica, se trabajó sobre el plugin JATS Parser, que permite generar automáticamente documentos PDF y HTML a partir de un XML JATS. El principal aporte aquí fue la incorporación de soporte para múltiples plantillas, que cada revista podrá definir según sus requerimientos. Posteriormente, se generó una primera plantilla básica a modo de ejemplo, que incorpora metadatos (obtenidos de OJS), el cuerpo del artículo y las referencias bibliográficas (obtenidos desde el XML).

A partir de esta mejora, no solo se facilita la generación automática de los distintos formatos, sino que además se potencia la personalización del diseño: cada revista, universidad o editorial puede crear y utilizar su propia plantilla, con lo cual se abren nuevas posibilidades para fortalecer la identidad editorial, asegurar la coherencia gráfica, y mantener estándares propios de forma sencilla y sostenible.



Imagen 1 - Plantilla de PDF utilizada en la UNLP.



Fuente: Villarreal y Ruiz (2025).

LIMITACIONES PARA CITACIÓN EN APA 7MA EDICIÓN Y VINCULACIÓN DE REFERENCIAS

Otro punto crítico detectado es el relacionado con la citación dentro del texto, especialmente bajo el estándar APA 7. El estándar XML JATS no define un estilo de citación particular, sino una estructura general. Sin embargo, al momento de generar un PDF (o cualquier otro formato), se debe aplicar un estilo concreto de citación de acuerdo a las normas editoriales de cada revista. Si bien el plugin JATSParser permite generar una lista de referencias en estilos como APA, IEEE y otros, no resuelve cómo deben mostrarse las citas dentro del cuerpo del artículo. En IEEE esto no es un problema, ya que cada referencia es citada con un número entre corchetes, como ser [1]. pero en APA, una misma referencia puede citarse de formas distintas según la redacción, por ejemplo:

Cita parentética: "El sol es una estrella (Juan, 2025)."

Cita narrativa: "Según Juan (2025), el sol es una estrella."

Cita directa con número de página: "... (Sánchez, 2020, p. 132)."



Cuando se marcan estas citas en el XML (por ejemplo, con Texture) y se vinculan con sus referencias el resultado es el siguiente:

El texto convertido presenta: "El sol es una estrella [1]."

O bien: "Según Juan [1], el sol es una estrella."

En consecuencia, al momento de generar el PDF desde el XML JATS, dado que solo se cuenta con un marcador del tipo [1], no es posible saber si se trata de una cita narrativa o parentética, lo que impide que el sistema aplique el formato correcto automáticamente. Por lo tanto, no es solo una cuestión de generación automática, sino de vincular correctamente las citas con sus referencias respetando las variaciones que impone cada estándar de citación. Se decidió entonces abordar estos casos con una solución pragmática: una interfaz que permita a los usuarios seleccionar manualmente el tipo de cita para cada referencia antes de generar el PDF. Si bien esta funcionalidad resolvió el problema en los casos de uso, quizás no sea una solución definitiva, sino un paso intermedio hasta llegar a una solución superadora. En esta solución, se sumó la posibilidad de guardar y reutilizar las configuraciones para facilitar el trabajo editorial cuando se trata de varios artículos o versiones, asegurando así una citación coherente a lo largo de distintas publicaciones.

CONCLUSIÓN Y PROYECTOS A FUTURO

En este trabajo se propone una estrategia para mejorar y optimizar los procesos de marcación y generación de XML JATS en revistas científicas, a través de la adaptación y extensión de herramientas existentes. A partir de estas modificaciones, se ha logrado no solo resolver problemas técnicos puntuales, sino también avanzar en una propuesta más amplia de autonomía editorial. Uno de los aportes centrales es que ahora los equipos editoriales tendrán la posibilidad de generar sus propios archivos XML y PDF sin depender de plataformas externas. Esto significa un paso importante hacia la independencia de los equipos editoriales, permitiendo que cada revista defina sus políticas y estilos de publicación sin estar condicionada por las decisiones o limitaciones de otros actores.

Actualmente, muchas revistas dependen de herramientas externas como Scielo Markup o Marcaly (versión para Redalyc o para AmeliCA) para la generación del XML JATS. Si bien estas herramientas han sido claves para democratizar el acceso a la marcación, también imponen restricciones tanto para acceder a su uso como para integrarse con los procesos editoriales de cada revista. Sumado a esto, el aumento constante de revistas que desean utilizar estos sistemas vuelve al modelo de software como servicio centralizado de estas instituciones difícil de sostener a mediano plazo. Con este desarrollo, se abre un nuevo camino que permite a las revistas ganar independencia, basado en un modelo descentralizado, autogestionado y de código abierto, que incluye tanto a revistas con amplia trayectoria en el uso de estas herramientas



como también a aquellas que desean comenzar a marcar sus artículos pero aún no “cumplen” con los requisitos impuestos para acceder a plataformas de terceros.

PROYECCIONES FUTURAS

A partir del desarrollo técnico y estratégico que se realizó en este trabajo, se abre un panorama de nuevas posibilidades para seguir avanzando en la autonomía y profesionalización del flujo editorial. Gracias a la decisión de extender la herramienta JATSParser incorporando una arquitectura que permite sumar plantillas personalizadas y configuraciones de citas, cualquier equipo podrá adaptar esta solución a sus propias revistas, estilos y políticas editoriales. Esta decisión estratégica sienta las bases para que en un futuro se puedan incorporar otros estilos de citación más allá de APA, como IEEE, Vancouver, Chicago, entre otros.

De igual manera, la estructura modular de plantillas desarrollada permite que se agreguen nuevos modelos de PDF, adaptados a las distintas revistas y requerimientos institucionales, sin necesidad de reescribir la herramienta desde cero. También es posible pensar en otros formatos de exportación, como HTML, ePUB, Mobi o TEI, ya que la herramienta propuesta no impone límites en los formatos de exportación. Esto incluso abre nuevos caminos en el desarrollo de soluciones para fomentar la accesibilidad web mediante formatos como ePUB. (De Giusti *et al.*, 2016).

No obstante, para que este enfoque pueda sostenerse y escalar a otros contextos, es fundamental avanzar también en la generación de documentación técnica clara y accesible, que permita a otros equipos implementar y extender estas funcionalidades sin depender directamente de quienes iniciamos el proyecto. Esta documentación deberá incluir tanto guías para la creación de nuevas plantillas como para el manejo de las configuraciones de citas, así como también criterios de buenas prácticas para su integración con el flujo editorial de OJS. Por último, se proyecta continuar y profundizar el vínculo con el equipo de PKP y la comunidad de desarrolladores de OJS, con el objetivo de aportar estas soluciones como contribuciones abiertas, que puedan ser discutidas, mejoradas y eventualmente integradas en versiones futuras de la herramienta, fortaleciendo la perspectiva de colaboración y construcción colectiva del ecosistema de publicación científica.

BIBLIOGRAFÍA

Becerril-García, A., Aguado-López, E. (2019). The End of a Centralized Open Access Project and the Beginning of a Community-Based Sustainable Infrastructure for Latin America. In L. Chan & P. Mounier (éds.), *Connecting the Knowledge Commons — From Projects to Sustainable Infrastructure* (1-). <https://doi.org/10.4000/books.oep.9003>



Becerril-García, A., Aguado López, E. y Macedo García, A. (2023). Marcalyc: software para la marcación XML JATS para las revistas científicas de acceso abierto diamante. *Palabra Clave (La Plata)*, 12(2), e179. <https://doi.org/10.24215/18539912e179>

Böschen, I. (2021). Software review: The JATSdecoder package—extract metadata, abstract and sectioned text from NISO-JATS coded XML documents; Insights to PubMed central's open access database. *Scientometrics*, 126, 9585–9601. <https://doi.org/10.1007/s11192-021-04162-z>

De Giusti, M. R., Lira, A. J., Rodríguez Vuan, J. P. & Villarreal, G. L. (2016). Accesibilidad de los contenidos en un repositorio institucional: análisis, herramientas y usos del formato EPUB. *E-Ciencias de la Información*, 6(2), 1–23. <https://doi.org/10.15517/eci.v6i2.23690>

McGlone, J. (2013). Preserving and publishing digital content using XML workflows. *Geneseo*: IDS Project Press, 97-108. <http://hdl.handle.net/2027.42/99563>

National Information Standards Organization. (2019). *ANSI/NISO Z39.96-2019, JATS: Journal Article Tag Suite, version 1.2*. <https://www.niso.org/standards/z3996-2019-jats>

Priem, J., Piwowar, H. & Orr, R. (2022). OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. ArXiv. <https://arxiv.org/pdf/2205.01833>

Redalyc. (2016a, 29 de julio). ¿Qué es XML?. *Blogredalyc*. <https://blogredalyc.wordpress.com/2016/07/29/que-es-xml/>

Redalyc. (2016b, 29 de julio). ¿Qué es JATS?. *Blogredalyc*. <https://blogredalyc.wordpress.com/2016/07/29/que-es-jats/>

Segundo, W., Matas, L., Nóbrega, T., Filho, J. E. S. & Mena-Chalco, J. (2023). dARK:a Decentralized Blockchain Implementation of ARK Persistent Identifiers. *EasyChair Preprints*, 9516. <https://easychair.org/publications/preprint/GpWB>

Soler, S., García, D., Villaruel, G. L. y Ruiz, A. (22-24 de octubre de 2024) Evaluación de estrategias de servicios de marcación y de publicación para artículos científicos [presentación en congreso]. *Conferencia Internacional BIREDIAL-ISTEC*. Santiago de Chile, Chile. <http://sedici.unlp.edu.ar/handle/10915/177616>

Velez-Estevez, A., Perez, I. J., García-Sánchez, P., Moral-Munoz, J. A. y Cobo, M. J. (2023). New trends in bibliometric APIs: A comparative analysis. *Information Processing & Management*, 60(4), 103385. <https://doi.org/10.1016/j.ipm.2023.103385>

Villarreal, G. L. y Ruiz, A. (2025, 19 de marzo). SUMARC: Sistema UNLP para el marcado de revistas científicas [conferencia principal]. *Presentación SUMARC*. <https://sedici.unlp.edu.ar/handle/10915/177809>



ANEXO 1

RESUMEN BIOGRÁFICO DE LOS/AS AUTORES/AS

Santiago Soler

Estudiante avanzado de la Licenciatura en Informática de la UNLP. Es desarrollador en PREBI-SEDICI, con enfoque en Open Journal Systems (OJS), y participa en proyectos de personalización y creación de plugins para optimizar la gestión de revistas científicas. Además, es investigador en nuevas tecnologías para mejorar los flujos de trabajo editorial. Está abocado a la promoción del avance científico a través del desarrollo de software de código abierto. ORCID: <https://orcid.org/0009-0003-0594-9581>

Tomás Messineo

Estudiante de la Licenciatura en Sistemas de la Universidad Nacional de La Plata (UNLP). Se incorporó recientemente al equipo de desarrollo de PREBI-SEDICI, donde participa activamente en el proyecto de marcación de artículos científicos de la UNLP. Su trabajo se enfoca en el desarrollo y mejora de herramientas orientadas al procesamiento y estandarización de publicaciones científicas, colaborando en la implementación de flujos de trabajo automatizados para la generación de XML JATS. ORCID: <https://orcid.org/0009-0007-9086-3609>

Dolores García

Licenciada en Comunicación Social, egresada de la Facultad de Periodismo y Comunicación Social de la UNLP en 2018. Fue becaria y pasante de la Comisión de Investigaciones Científicas de la provincia de Buenos Aires. Desde 2016 desarrolla tareas de gestión, técnicas y capacitaciones a equipos editoriales y a comités en el Portal de Revistas de la UNLP y en el Portal de Congresos de la UNLP. Se desempeña como asistente técnica en la Revista Argentina de Antropología Biológica y en la Asociación Argentina de Antropología Biológica. Forma parte del equipo de RevPsi como asistente editorial. En abril de 2023, asumió el cargo de Profesional Adjunto en CESGI, otorgado por la Comisión de Investigaciones Científicas de la provincia de Buenos Aires. ORCID: <https://orcid.org/0000-0002-6686-3138>

Gonzalo Luján Villarreal

Doctor en Ciencias Informáticas, es director de PREBI-SEDICI de la Universidad Nacional de La Plata, director del Centro de Servicios en Gestión de Información (CESGI, 2016) de la Comisión



de Investigaciones Científicas de la Provincia de Buenos Aires, coordinador informático de revistas científicas de la Universidad Nacional de La Plata y profesor de la Facultad de Informática de la misma universidad. ORCID: <https://orcid.org/0000-0002-3602-8211>

Adela Ruiz

Licenciada en Comunicación Social por la Universidad Nacional de La Plata (UNLP) y Diplomada en Políticas Editoriales y Proyecto Cultural por la Universidad de Buenos Aires (UBA). Desde 2021, se desempeña como Coordinadora General de Revistas Científicas de la UNLP y, desde 2014, como Directora de Publicaciones Científicas de la Facultad de Periodismo y Comunicación Social (FPyCS) y como asesora en Sistemas de Evaluación de Revistas de la Facultad de Artes (FA). Es profesora titular del Taller de Edición Técnica y docente de la Especialización en Edición de la FPyCS. ORCID: <https://orcid.org/0000-0002-2873-006X>