



Facultad de Informática
Universidad Nacional de La Plata

Integrando repositorios digitales de recursos educativos abiertos con plataformas virtuales de aprendizaje

Tesina de Licenciatura en Informática

Alumna: María Emilia Charnelli

Directoras: María Alejandra Schiavoni - Ana Paola Amadeo

20 de febrero de 2014



El presente trabajo se hizo en el marco de la Beca de Entrenamiento BENTR12 otorgada por la Comisión de Investigaciones Científicas (CIC) en octubre de 2012 y se llevó a cabo en el Laboratorio de Investigación en Nuevas Tecnologías Informáticas (LINTI) de nuestra Facultad.

Director de Beca: Javier Diaz.



El objetivo de este trabajo es facilitar la publicación de las producciones realizadas por los alumnos en el marco de las cátedras, en un repositorio digital de acceso abierto, con el fin de otorgarles mayor visibilidad y difusión.



- El acceso a materiales educativos digitales se limita generalmente a los alumnos matriculados en un curso. El movimiento de **Recursos Educativos Abiertos (OER)** tiene como objetivo romper las barreras y fomentar y permitir libremente el intercambio de contenidos.
- La integración de **plataformas virtuales de aprendizaje** con otras plataformas y aplicaciones puede ayudar a aumentar la disponibilidad de recursos y la comunicación entre los usuarios.
- **Repositorios digitales** permiten la preservación y reutilización de contenido, acceso permanente, facilidad de búsqueda y recuperación.



- En la Facultad de Informática se vienen usando entornos de enseñanza y aprendizaje desde hace más de nueve años: **WebUNLP, Moodle**.
- Las carreras de nuestra Facultad cuentan con numerosas materias de índole netamente práctico. La posibilidad de publicar en un repositorio digital los proyectos realizados por los alumnos aumenta la disponibilidad de estos recursos para que puedan ser aprovechados por otras personas y de esta manera generar nuevas experiencias de aprendizaje.
- La publicación del material por parte del docente requiere de un tiempo considerable por las especificaciones requeridas.
- El trabajo desarrollado en esta tesina forma parte de un proyecto más amplio que se viene llevando a cabo en el LINTI. Integración de la plataforma con redes sociales, sistema de gestión de alumnos.



Para la integración propuesta se trabajó sobre la plataforma virtual Moodle y un repositorio digital DSpace.



- Entre las plataformas de la Facultad se eligió **Moodle** por ser *open source*, gran comunidad de usuarios y desarrollados. Se utilizaron las versiones 2.0, 2.2 y 2.4.
- Se eligió **DSpace** por ser *open source*, fácil de instalar y usar, comunidad activa, más usado a nivel mundial. Se utilizó la versión 3.1

Para la comunicación se utilizó el protocolo **SWORD**.



- La Facultad viene trabajando con Moodle desde hace aproximadamente 9 años como complemento de las clases presenciales de cursos de las carreras de grado, de postgrado y de la Secretaría de Extensión.
- Actualmente en la versión 2.4.
- La plataforma incluye más de 10000 usuarios registrados en alrededor de 170 cursos.
- Al ser un sistema de software libre permite extender funcionalidades a través de actividades o bloques.

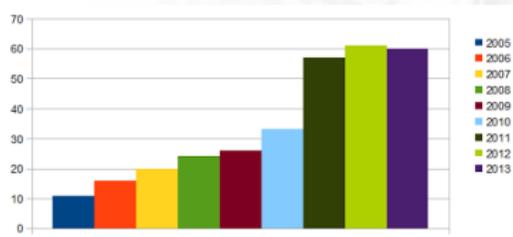


Figura : Cantidad de cursos por año

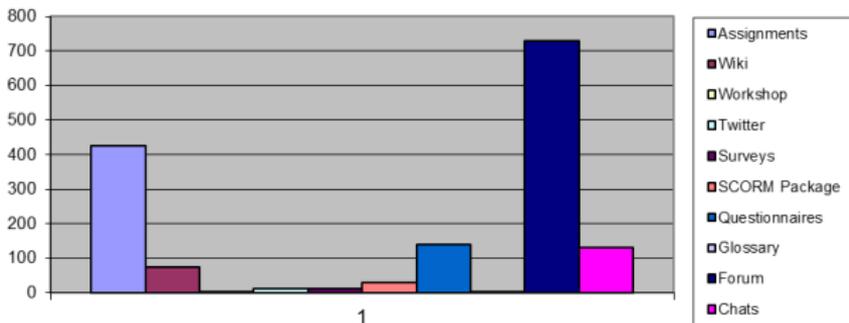


Figura : Actividades utilizadas en la plataforma virtual Moodle

- Las Tareas o Assignments ocupan el segundo lugar, donde los recursos incluidos en ellas son el eje central y motivador de esta tesina.



- Hay diferentes tipos de Tarea: un solo archivo, varios archivos, texto en línea, no en línea.
- El docente puede ver el listado de todas las entregas realizadas para una determinada tarea, ver/modificar la calificación y las observaciones.
- Los alumnos sólo pueden ver sus entregas.
- La herramienta propuesta permite extender la funcionalidad del módulo Tarea para que el docente pueda exportar las entregas realizadas por los alumnos en forma semi automática.



► curso 1 ► General ► Entrega N°1 ► Ver 2 tareas enviadas

[Ver todas las calificaciones del curso](#)

Nombre : **Todos**ABCDEFGHIJKLMNÑOPQRSTUVWXYZ

Apellido(s) : **Todos**ABCDEFGHIJKLMNÑOPQRSTUVWXYZ

[Descargar todas las tareas en un zip](#)

 Nombre / Apellido(s) ↓	Dirección de correo	Calificación	Comentario	Última modificación (
 Paola Amadeo	amadeo@a.com	-		 pamadeo.zip Monday, 17 de Februar, 2014, 15:10
 María Emilia Charnelli	mcharnelli@a.com	-		
 Alejandra Schiavoni	aschiavoni@a.com	-		 aschiavoni.zip Monday, 17 de Februar, 2014, 15:09

Figura : Vista de las entregas de una tarea en Moodle



Con el fin de tener una herramienta flexible y que pueda ser incorporada fácilmente en cualquier entorno Moodle, se decidió crear un módulo con los siguientes funcionalidades:

- 1 Recuperar los archivos de cada entrega
- 2 Completar automáticamente metadatos asociados a la tarea
- 3 Generar un nuevo paquete con los recursos y los metadatos
- 4 Enviar el recurso empaquetado a un repositorio asociado



- Definir pautas de organización para estandarizar la estructura de los recursos.
- Establecer criterios de clasificación o “marcado” automático de los recursos.



Recursos uniformes claramente especificados y clasificados



Moodle permite recuperar información de dos contextos diferentes: el curso dentro de la plataforma y la tarea creada para la entrega.

- **Contexto del Curso** nombre de la materia, nombre de los docentes, direcciones de contacto, año de cursada, etc.
- **Contexto de la Tarea** fecha de entrega, descripción, nombre del alumno, archivo entregado, tipo de archivo, correo electrónico.



- La elección de los metadatos a utilizar es una tarea delicada y que involucra conocimientos de gente experta en el tema.
- En una primera instancia, se realizaron entrevistas con los docentes responsables de las asignaturas en las cuales se realizan entregas a través de Moodle.
- Se identificaron metadatos adicionales: autores de un grupo, título del trabajo, requerimientos de hardware y software que serían proporcionados por los alumnos a través de archivos adicionales.
- Se utilizó el estándar de metadatos Dublin Core utilizado por DSpace.



Los recursos publicados en el repositorio tendrían los siguientes metadatos:

Metadato	Modelo de Metadato	Origen
Título	Dublin Core	<i>Contexto del Curso / Tarea</i>
Autor	Dublin Core	
Descripción	Dublin Core	
Tutor	Dublin Core	
Formato de archivo	Dublin Core	
Fecha	Dublin Core	
Contacto del docente	propio	
Contacto del alumno	propio	
Resumen	Dublin Core	<i>Información adicional</i>
Lenguaje	propio	
Sistema Operativo	propio	
Editor	Dublin Core	<i>Por defecto</i>
Licencia	Dublin Core	
Fuente	Dublin Core	
Idioma	Dublin Core	
Tipo de recurso	Dublin Core	



- El módulo recopila la información de la tarea: archivos propios y archivos con información adicional
- Además puede incorporar información por defecto como la Licencia del recurso, el idioma español. Previamente configurado por el docente.

Una vez publicados...

- Los bibliotecarios podrían completar los metadatos relacionados (Relation), Cobertura y Contribuyente.
- Realizar catalogaciones adicionales.



Recursos con los 15 metadatos básicos DC, con una carga manual mínima, y a través de distintos actores: alumnos, docentes y bibliotecarios.



La comunicación entre Moodle y DSpace se detalla de la siguiente manera:

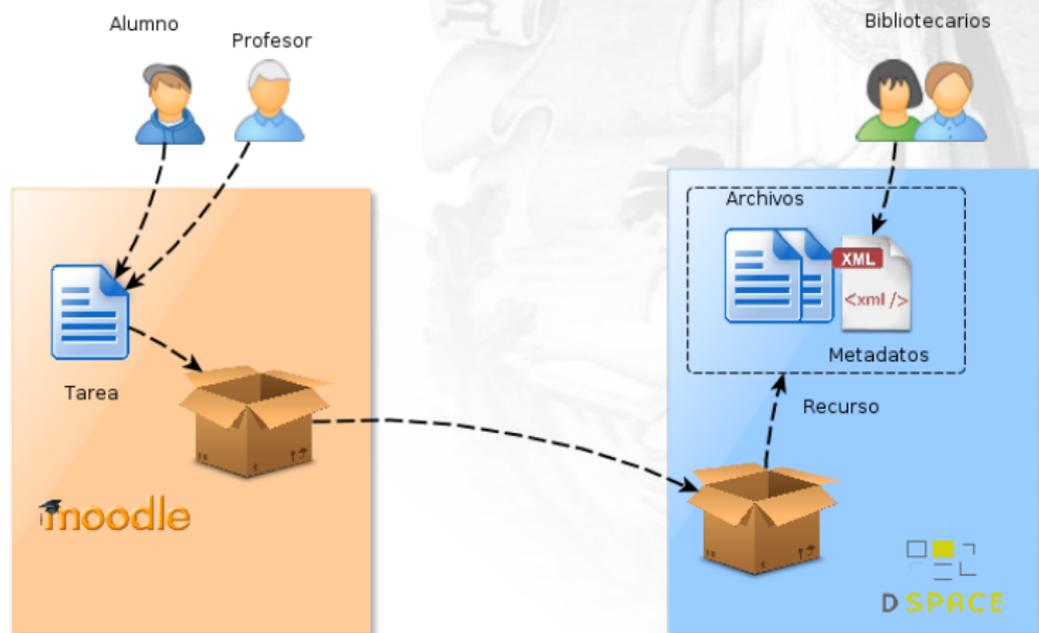


Figura : Esquema de comunicación entre Moodle y DSpace a través de SWORD



- 1 El alumno realiza una entrega sobre una tarea específica
- 2 El Docente envía el trabajo al repositorio.
- 3 El módulo toma cada una de las entregas seleccionadas para el envío, y de cada una de ellas toma todos sus archivos. Luego empaqueta los archivos y los metadatos en un archivo zip.
- 4 El paquete se envía a través de SWORD al repositorio indicado.
- 5 El módulo SWORD instalado por defecto en DSpace desarma el paquete y recupera los metadatos y los archivos asociados para crear un nuevo recurso.
- 6 El módulo SWORD de DSpace envía una respuesta en formato xml al módulo creado en Moodle.



- La carga de metadatos se divide en dos etapas principales:
 - Cargar automáticamente los metadatos a través del contexto de la tarea y del curso
 - Cargar automáticamente los metadatos a través de archivos adicionales
- Desde la aplicación se genera un paquete METS con los archivos y los metadatos descriptivos, administrativos y estructurales que luego usará el repositorio para la incorporación del recurso.
- Luego, este conjunto de archivos se integra en un único archivo zip que será enviado a través del protocolo al repositorio especificado.



El módulo se divide en dos interfaces

- **Interfaz de configuración**

En la parte de configuración el Docente podrá:

- Asociar el repositorio al que quiera poder exportar los trabajos
- Establecer metadato/s por defecto

Actualizando SWORD ⓘ

General

nombre*

Información del repositorio

url del repositorio*

usuario*

contraseña*

Valores de los metadatos por ▶ Curso1 ▶ General ▶ Entrega Nº1

Palabras claves

Derechos

Idioma

Publicador

Ver todas

Nombre : TodosABCDEFGHIJKLMNÑOPQRSTUVWXYZ

Apellido : TodosABCDEFGHIJKLMNÑOPQRSTUVWXYZ

Enviar al Repositorio

Estado	Seleccionar	Nombre / Apellido	D
No enviado	<input type="checkbox"/>	Paola Amadeo	par
No enviado	<input type="checkbox"/>	Maria Emilia Charnelli	me

- **Interfaz de envío**

En la interfaz de envío el docente podrá:

- Seleccionar y enviar tareas
- Ver los estados de cada una de las entregas: enviada o no.

Desarrollo de la Herramienta de Integración

Protocolo SWORD (Simple Web-service Offering Repository Deposit)



- Aplicación que define de forma sencilla el depósito remoto de contenidos en un repositorio desde otras aplicaciones.
- A partir de la versión v2 se añaden operaciones CRUD.
- La implementación de la librería de SWORD utiliza el estándar METS.
- Disponibilidad de librerías en diversos lenguajes como PHP, Java, Python y Ruby.
- Dspace, Fedora Eprints, IntraLibrary, DataBank implementan este servicio.



- La librería SWORD permite:
 - depositar un recurso a un repositorio
 - crear el documento xml con los metadatos del recurso
 - crear el paquete con los archivos y el documento xml para poder ser exportado
- La clase SWORDAPPClient provee funcionalidades para crear un cliente SWORD. Funciones principales: *servicedocument* y *deposit*.
- La versión v2 incluye otras funciones importantes para completar un depósito incompleto, para reemplazar el contenido, reemplazar metadatos o para agregar un archivo a un recurso ya publicado.



Se implementó una clase principal **sword_lib** que define las funcionalidades importantes para que se puedan depositar entregas de los alumnos en el repositorio. **sword_submissions** es la función principal.

```
public function sword_submissions($cm,$course,$assignment ,
    $submissions_id) {
    (...)
    foreach ($select_submissions as $submission) {
        $a_userid = $submission->userid; //get userid

        foreach ($files as $file) {
            if($file) {
                if($file->get_filename()=="*.txt"){
                    $contents = $file->get_content();
                    $arr = explode("\n", $contents);
                }
                ...
                $this->copyFileToTemp($file);
                $package = $this->makePackage($file, $arr);
                $this->sendToRepository($paquete);
            }
        }
    }
}
```



En primera instancia el depósito remoto se pensó como una funcionalidad extra del módulo de Tareas.

- Se modificó el código del módulo Tareas -por defecto en Moodle-
- Se modificaron las tablas de la base de datos para las tareas
- Se agregaron nuevas clases y librerías.



No representaba una solución que fuera escalable e independiente.



Creación de un **módulo independiente** que se basa en mayor parte del módulo Tarea añadiendo la funcionalidad de envío.

Módulo Tarea extendido



La comunidad de Moodle dispone de plantillas para poder crear actividades o bloques de forma sencilla y que sean fáciles de incorporar por cualquier usuario. Las partes más importantes del módulo son:

- Archivo **view.php**
Pantalla principal que se muestra en una instancia de la actividad
- Carpeta **db**
Se agregan todas las funciones que se deben llevar a cabo para cuando se instala, desinstala y se actualiza un módulo. También se encarga de crear las tablas y campos necesarios en la base de datos de nuestra actividad.
- Archivo **lib.php**
En esta carpeta va toda la funcionalidad del módulo.



- Se extendió de la clase principal del módulo de Tareas **assignment_base**, para evitar tener el menor código repetido.
- A la visualización de las tareas se incorporaron campos nuevos para poder seleccionar tareas y para poder ver el estado de envío de las mismas.
- Finalmente se incorporaron más pantallas, una pantalla inicial donde el docente ve el listado de las tareas llevadas a cabo en un curso y otra para la parte de configuración de la misma.
- La función **sword_submissions** mencionada anteriormente recibe el curso, la tarea, y las entregas seleccionadas por el docente, y a partir de ahí, genera el paquete zip para su posterior envío.



Para la versión primera de este módulo se crearon las siguientes tablas:

- **sword** La tabla sword registra cada una de las instancias del módulo creadas. Guarda una referencia al id del curso y guarda información del repositorio: url, usuario, contraseña. A su vez, incorpora metadatos que se pueden establecer por defecto: *subject*, *rights*, *language*, *publisher*.
- **sword_submissions** La tabla sword_submissions registra todas las entregas que fueron enviados o fallaron en el envío .



En la pantalla inicial, se muestra el listado de tareas creadas en ese curso.

The screenshot shows a web interface for a course named 'Curso1'. At the top, there is a breadcrumb trail: 'Página Principal > Mis cursos > Curso1 > General > sword'. Below this is a 'Navegación' (Navigation) sidebar with a tree structure. The main content area is titled 'Lista de Tareas' and contains a table with a header 'Tarea' and three rows of tasks: 'Entrega N°2', 'Entrega N°1', and 'Entrega N°3'.

Curso1

[Página Principal](#) > [Mis cursos](#) > [Curso1](#) > [General](#) > [sword](#)

Navegación

- [Página Principal](#)
- [Área personal](#)
- ▶ [Páginas del sitio](#)
- ▶ [Mi perfil](#)
- ▼ [Mis cursos](#)
 - ▼ [Curso1](#)
 - ▶ [Participantes](#)
 - ▶ [Informes](#)
 - ▼ [General](#)
 - [Novedades](#)
 - [Tarea1](#)

Lista de Tareas

Tarea
Entrega N°2
Entrega N°1
Entrega N°3

Figura : Vista principal del módulo Tareas



ds ▶ Curso1 ▶ General ▶ Entrega N°1

[Ver todas las calificaciones del curso](#)

Nombre : **Todos**ABCDEFGHIJKLMNÑOPQRSTUVWXYZ
Apellido : **Todos**ABCDEFGHIJKLMNÑOPQRSTUVWXYZ

Enviar al Repositorio

Estado	Seleccionar	Nombre / Apellido ↓	Dirección de correo	Cal
No enviado	<input type="checkbox"/>	 Paola Amadeo	pamadeo@gmail.com	-
No enviado	<input type="checkbox"/>	 María Emilia Charnelli	mecharnelli@gmail.com	-
No enviado	<input type="checkbox"/>	 Alejandra Schiavoni	aschiavoni@gmail.com	-
No enviado	<input type="checkbox"/>	 Admin Usuario	a@a.com	-

Figura : Pantalla donde se eligen las tareas a publicar



Ver todas las calificaciones del curso

Nombre : **Todos** ABCDEFGHIJKLMNOPQRSTUVWXYZ
Apellido : **Todos** ABCDEFGHIJKLMNOPQRSTUVWXYZ

Enviar al Repositorio

Estado	Seleccionar	Nombre / Apellido ↓	Dirección de correo	Calificación	Comentario	Última
Enviado	<input type="checkbox"/>	Paola Amadeo	pamadeo@gmail.com	-		pam jueves, 9
No enviado	<input type="checkbox"/>	María Emilia Charnelli	mecharnelli@gmail.com	-		
Enviado	<input type="checkbox"/>	Alejandra Schiavoni	aschiavoni@gmail.com	-		asch jueves, 9
No enviado	<input checked="" type="checkbox"/>	Admin Usuario	a@a.com	-		prue jueves, 9 10:08 (1

Figura : Envío de una entrega seleccionada



📦 Actualizando SWORD ?

General

nombre* ?

Información del repositorio

url del repositorio*

usuario*

contraseña*

Valores de los metadatos por defecto

Palabras claves

Derechos

Idioma

Publicador

Figura : Pantalla para la creación/edición de una instancia de SWORD



- Se instaló el módulo desarrollado en el servidor de pruebas del Moodle catedras.info.unlp.edu.ar
- El módulo deposita correctamente cada una de las entregas que hayan sido seleccionadas para ser enviadas.
- Una vez enviado al repositorio, el docente puede consultar la tarea desde la plataforma o en el mismo repositorio.



Comunidades y colecciones



Por fecha de publicación



Autores



Títulos



Temas

ESTA COMUNIDAD



Por fecha de publicación

MATERIAL DE CÁTEDRAS

SUBCOMUNIDADES EN ESTA COMUNIDAD

Biblioteca FCE

Diseño Centrado en el Usuario

Laboratorio de Software

Seminario de Lenguajes C

Seminario de Lenguajes Python

Taller de Tecnologías de Producción de Software - Opción Java

Tecnologías Aplicadas a BI

ENVÍOS RECIENTES



Comunidades y colecciones

Por fecha de publicación

Autores

Títulos

Temas

ESTA COLECCIÓN

Por fecha de publicación

Autores

Títulos

Temas

ACCEDER

ADMINISTRACIÓN DE HISTORIAS CLÍNICAS PARA EL HIAEP

TÍTULO: Administración de Historias Clínicas para el HIAEP

AUTOR: Nola, Eduardo; DiGrazia, Anabella

RESUMEN: Implementar un sistema que permita mantener actualizada y accesible la Historia Social de los niños atendidos por las trabajadoras sociales del Hospital de Niños de La Plata. Los profesionales del Servicio Social del HIAEP Sor María Ludovica, que atienden más de 60.000 niños por año, mantienen toda esta información en extensos formularios de papel, lo que provoca previsible dificultades en cuanto a su actualización, su acceso, etc.

URI: <http://dspace.linti.unlp.edu.ar/handle/123456789/147>

FECHA: 2013-06-28

LICENCIA: BY-SA

FICHEROS EN EL ÍTEM



Consideraciones ...
torias Clínicas-
HIAEP.pdf



Guarda



Guardar

14.18Kb PDF



Sistema Informá ... l
de Niños de La
Plata.pdf



Guarda



Guardar

187.3Kb PDF

Descripción
del trabajo
presentado
en EST 2012
en el marco
de las JAIIO
2012



HospitalDeNiños.war



Guarda



Guardar

20.56Mb Desconocido

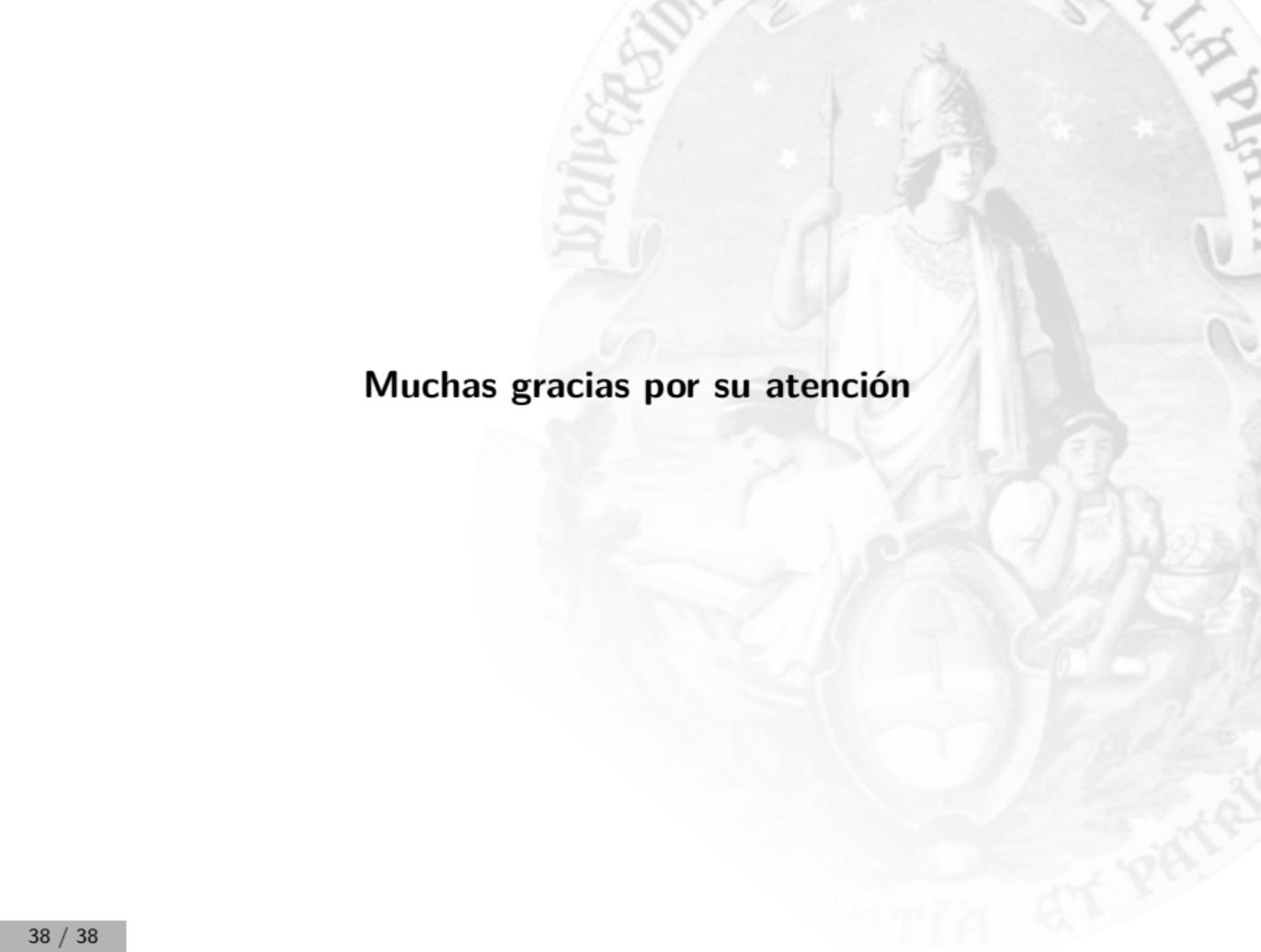
Sistema
desarrollado



- Este trabajo fue publicado en congresos nacionales e internacionales: WICC2013, INTED2013, IADIS2013 y CLEI2013.
- El módulo desarrollado facilita el intercambio y reutilización de recursos educativos que han sido desarrollados y almacenados en plataformas y herramientas heterogéneas.
- Este módulo permite reducir la sobrecarga que representa almacenar los mismos recursos en distintas plataformas.
- El módulo trabaja sin necesidad de la intervención explícita de un usuario para completar los metadatos elementales.
- Con sólo adicionar unos muy pocos metadatos en forma manual permitirán completar los 15 metadatos elementales de un recurso bibliográfico.



- SWORD V2 incorpora operaciones CRUD lo que permitiría poder eliminar o actualizar un recurso.
- Publicación de otro tipo de material contenido en la plataforma, como material teórico o práctico, apuntes, etc.
- Incorporar minería de texto para extraer palabras claves de los archivos de cada tarea.
- Extender la herramienta para que se pueda comunicar con repositorios que utilicen otro modelo de metadatos.



Muchas gracias por su atención