SIU SISTEMA DE INFORMACIÓN UNIVERSITARIA

CiN Consejo Interuniversitario Nacional

Curso de Koha

Orientado al personal de tecnología de las Universidades Nacionales

Jornada 4/5

Capacitadores:

- Pablo Bianchi
  Consultor
  pablo.bianchi@gmail.com
- Álvaro Hernán Gómez Cardozo
  (agomezcardozo@herrera.unt.edu.ar)
  UNT
- Claudio M. Fuhr
  (claudiofuhr@cab.cnea.gov.ar)
  Instituto Balseiro - UNCuyo - CNEA

30 de Septiembre de 2019 al 4 de Octubre 2019 - Ciudad Autónoma de Buenos Aires

bibliotecas@siu.edu.ar / https://www.siu.edu.ar/siu-bibliotecas/

Esta obra está bajo una Licencia Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional.

La actividad “Curso de capacitación en Koha para el personal de tecnología de las Universidades Nacionales” es posible gracias a un subsidio otorgado por la **Fundación Williams** al **Consejo Interuniversitario Nacional (CIN)** en el mes de Julio de 2019. La información del proyecto se encuentra en:

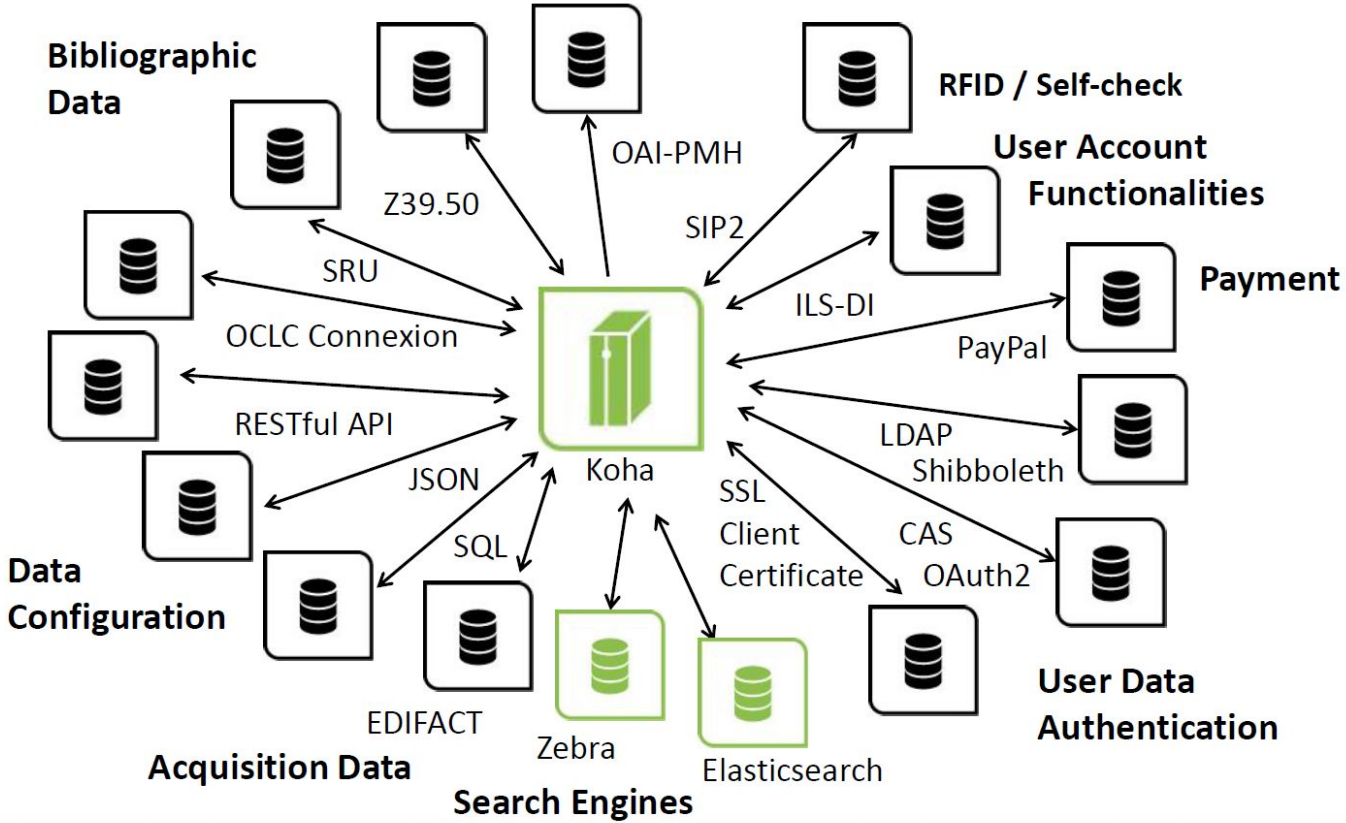
https://portal.comunidad.siu.edu.ar/novedades/noticias/202-curso-de-koha-para-personal-de-tecnologia-de-las-universidades-nacionales#becas

Los ejercicios presentados en la capacitación se han planteado de modo de simular requerimientos con el único objetivo de facilitar el proceso de aprendizaje de la plataforma KOHA. El planteo de los ejercicios no está implicando un compromiso por parte del SIU-CIN en el desarrollo, evaluación o implementación de los mismos para el Sistema Universitario. En los ejercicios en que se mencionan organizaciones reales, los casos planteados son simulaciones que no tienen necesariamente correlato con situaciones o requerimientos existentes.

Introducción

APIs and Protocols

BSZ Bibliotheksservice-Zentrum Baden-Württemberg



Fuente: https://es.scribd.com/document/418070477/Advanced-Koha-Workshop-katrin

Exposiciones relacionadas con la interoperabilidad de KOHA

INTEROPERABILIDAD CON KOHA

UNSE
Universidad Nacional
de Santiago del Estero

Biblioteca Central

APLICACIONES DE INTERFAZ

Ing. Bailon Santiago
Coordinador
Red de Bibliotecas UNSE
bailon.santiago@gmail.com

IMPLEMENTACIONES EN UNSE

- **Pagina WEB - CMS**

http://libnet.unse.edu.ar

- **Catálogo en Línea**

http://opac.unse.edu.ar:8082/

- **Repositorio Digital de Biblioteca Central**

http://bcdigi.unse.edu.ar:8080/jspui/

- **Proyecto Voluntariado BiblioRedSE**

Catálogo Colectivo Bibliotecas Populares + Bibliotecas de la Provincia [VuFind]

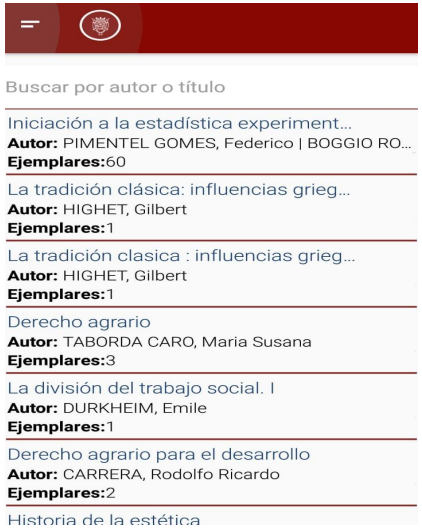
KOHA

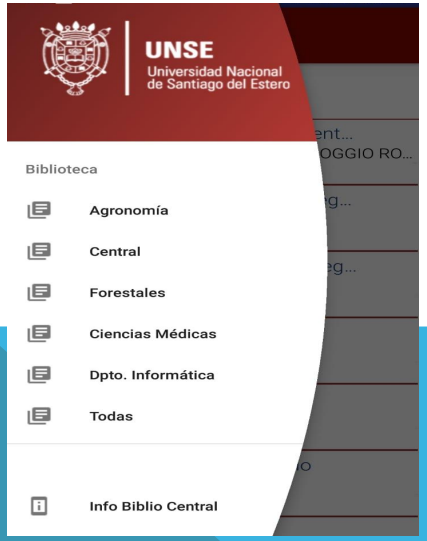
- Cantidad de Bibliotecas incorporadas: 7
- 5 Biblioteca en procesamiento
- Cantidad de Item aprox.: 10mil
- Cantidad de Socios: 825
- Cantidad de Prestamos 2019: 2000

DATOS SERVER

- Intel Core i5
- 8gb Ram
- 2 disco de 2TB en RAID 1
- S.O. Debian Jessie

INTERFAZ 1: CATALOGO





Objetivos:

- Acceso instantáneo al catalogo.
- Filtrar por Bibliotecas Fácilmente.
- Ver disponibilidad de item en las Sedes.

Desarrollo en conjunto con alumnos becados del Departamento de Informática

Año 2017

TABLAS QUE INTERVIENEN

biblio	[table]
biblionumber	int[10]
frameworkcode	varchar[4]
author	longtext[2147483647]
title	longtext[2147483647]
unititle	longtext[2147483647]
notes	longtext[2147483647]
serial	bit[0]
seriestitle	longtext[2147483647]
copyrightdate	smallint[5]
timestamp	timestamp[19]
datecreated	date[10]
abstract	longtext[2147483647]
< 0	0 rows 14 >

biblioitems	[table]
biblioitemnumber	int[10]
biblionumber	int[10]
volume	longtext[2147483647]
number	longtext[2147483647]
itemtype	varchar[10]
isbn	longtext[2147483647]
issn	longtext[2147483647]
ean	longtext[2147483647]
publicationyear	mediumtext[16777215]
publishercode	varchar[255]
volumedate	date[10]
volumedesc	mediumtext[16777215]
collectiontitle	longtext[2147483647]
collectionissn	mediumtext[16777215]
collectionvolume	longtext[2147483647]
editionstatement	mediumtext[16777215]
editionresponsibility	mediumtext[16777215]
timestamp	timestamp[19]
illus	varchar[255]
pages	varchar[255]
notes	longtext[2147483647]
size	varchar[255]
place	varchar[255]
lccn	varchar[25]
url	mediumtext[16777215]
cn_source	varchar[10]
cn_class	varchar[30]
cn_item	varchar[10]
cn_suffix	varchar[10]

biblioimages	[table]
imagenumber	int[10]
biblionumber	int[10]
mimetype	varchar[15]
imagefile	mediumblob[16777215]
thumbnail	mediumblob[16777215]
< 1	0 rows 0 >

branches	[table]
branchcode	varchar[10]
branchname	longtext[2147483647]
branchaddress1	longtext[2147483647]
branchaddress2	longtext[2147483647]
branchaddress3	longtext[2147483647]
branchzip	varchar[25]
branchcity	longtext[2147483647]
branchstate	longtext[2147483647]
branchcountry	mediumtext[16777215]
branchphone	longtext[2147483647]
branchfax	longtext[2147483647]
branchemail	longtext[2147483647]
branchreplyto	longtext[2147483647]
branchreturnpath	longtext[2147483647]
branchurl	longtext[2147483647]
issuing	tinyint[3]
branchip	varchar[15]
branchprinter	varchar[100]
branchnotes	longtext[2147483647]
opac_info	mediumtext[16777215]
geolocation	varchar[255]
marcorgcode	varchar[16]
pickup_location	bit[0]
< 0	0 rows 31 >

EJEMPLO DE WEB SERVICE CON JSON EN KOHA

```
<?php
 $param = $_POST['param'];
 $conexion = mysql_connect("localhost", "koha_app", “xxxxx") or die("problemas conexion");

 mysql_select_db("koha_xxxxxx",$conexion);

     $query="select b.author,b.title,b.biblionumber,b.notes,biblioi.isbn,biblioi.pages,branch.branchname,i.onloan,i.itemnumber
     from biblio b left outer join biblioitems biblioi on b.biblionumber = biblioi.biblionumber join items i on b.biblionumber =
i.biblionumber
     join branches branch on branch.branchcode = i.homebranch where i.biblionumber='".$param."'";

 $result = mysql_query($query) or die(mysql_error($conexion));
 while ($e = mysql_fetch_assoc($result))
 {
      $output[]=$e;/*array_map('utf8_encode',$e);*/
 }
 print(json_encode($output));
```

EJEMPLO BUSQUEDA EJEMPLAR

Buscar por autor o título

sql

MySQL Avanzado
Autor: SHUWARTZ, Baron
Ejemplares:2

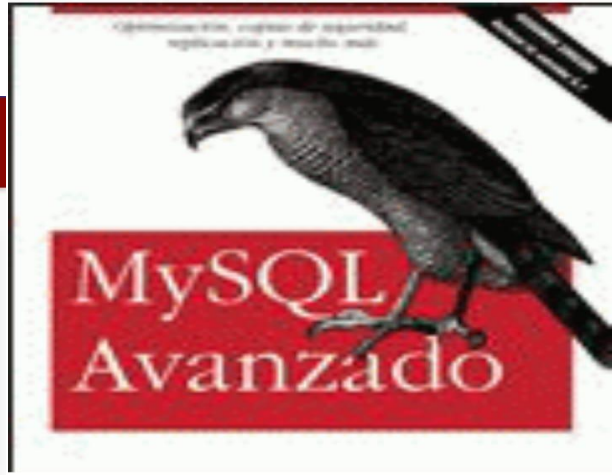
Microft sql server 2008 - Business Int...
Autor: VEERMAN, Erik
Ejemplares:3

SQL y JAVA. Guia para SWLJ, JDBC y...
Autor: MELTON, Jim
Ejemplares:1

SQL Server 2005, Designing a Databa...
Autor: MACKIN, J.C.
Ejemplares:1

Introducción al SQL para usuarios y p...
Autor: RIVERO CORNELIO, Enrique
Ejemplares:1

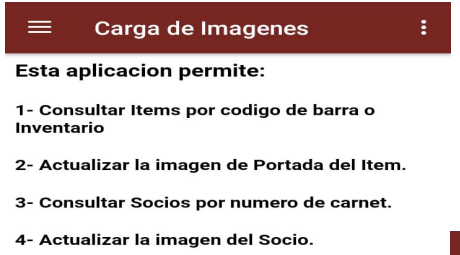
Optimizing and Maintaining a Databa...
Autor: THOMAS, Orin
Ejemplares:1

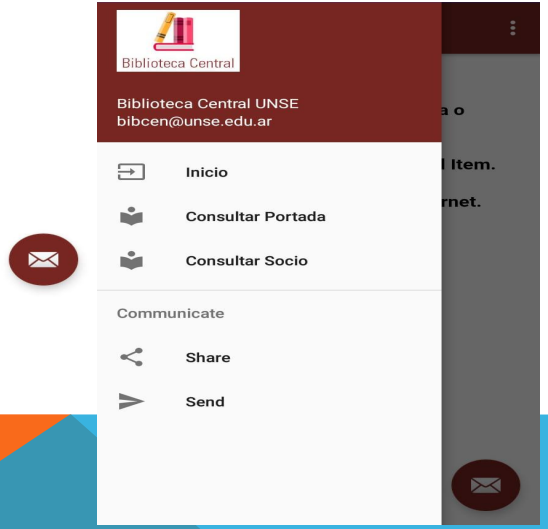


MYSQL AVANZADO

Autor	SHUWARTZ, Baron
Páginas:	832 p.
Biblioteca:	Biblioteca Central
Isbn:	978-84-415-2507-8
Estado	Disponible

INTERFAZ 2: CARGA DE IMÁGENES





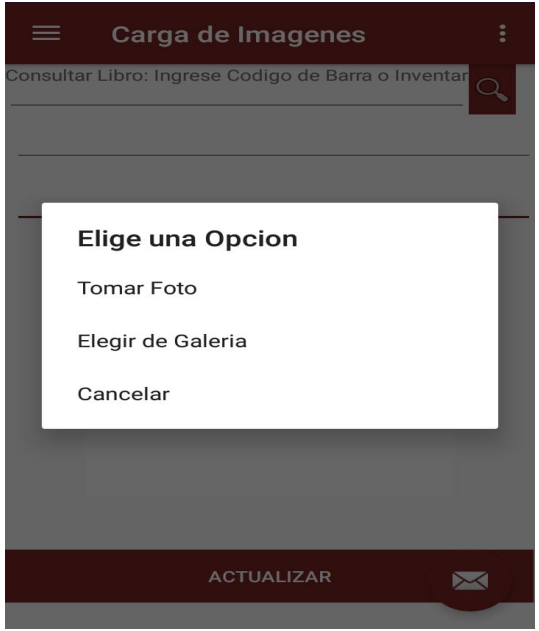
Objetivos:

- Consulta de Socio y ejemplar.
- Carga de Imagen de Socio y Ejemplar.

Desarrollado por Biblioteca Central UNSE

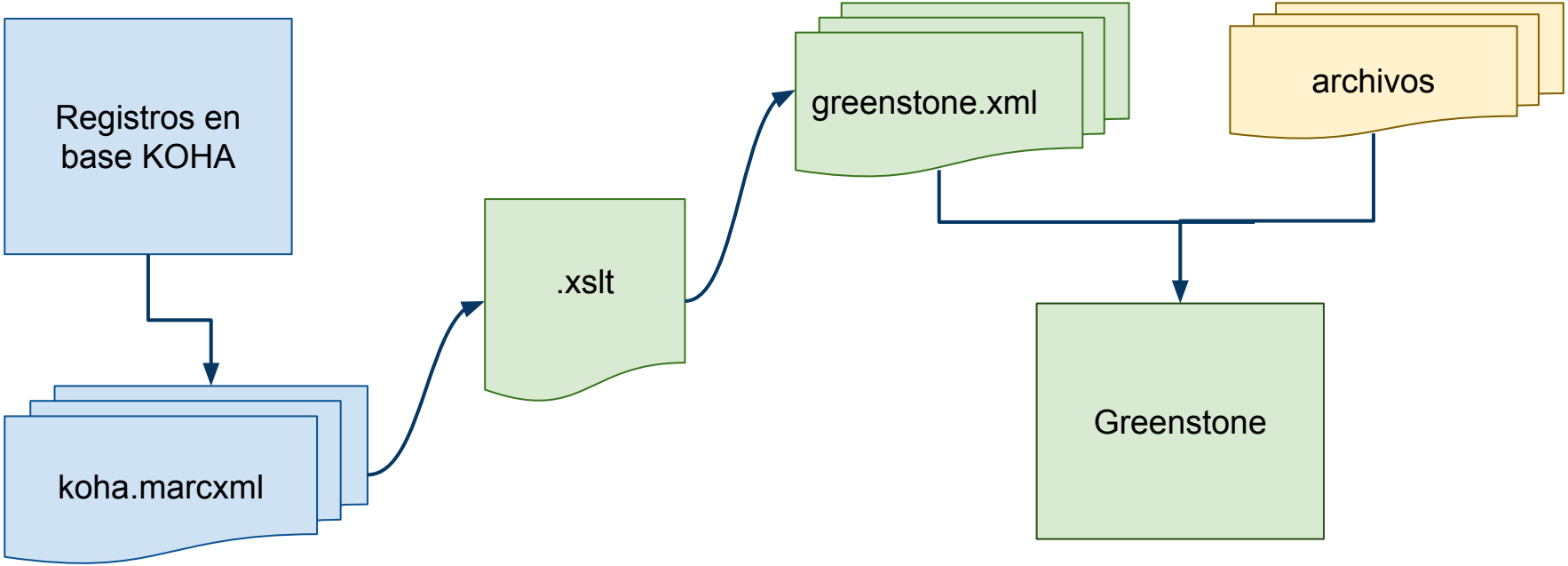
Año 2018

CARGA/CONSULTA DE IMÁGENES



Koha - Greenstone

Importación de registros en Greenstone desde Koha



Esquema sobre la importación KOHA-Greenstone

Estructura xml de archivos de exportación .marcxml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<collection>
    <record>
            <leader>02174nab a2200349ua 4500</leader>
            <controlfield tag="001">ART975</controlfield>
            <datafield tag="100">
                    <subfield code="a">Cloquell, Silvia</subfield>
            </datafield>
              ...
            <datafield tag="245">
                    <subfield code="a">Abordajes y enfoques acerca de la (...) rurales</subfield>
            </datafield>
    </record>
      ...
    <record>
            <leader>02981nmm a2200589ua 4500</leader>
            <controlfield tag="001">TDOC001501/1209</controlfield>
            <datafield tag="100">
              <subfield code="a">Gorrochategui,Nora L.</subfield>
            </datafield>
              ...
            <datafield tag="245">
              <subfield code="a">Cumplimiento del principio 10 del (...) en la Argentina</subfield>
             </datafield>
    </record>
</collection>
```

Estructura xml de archivos de importación .xml

```xml
<?xml version="1.0"?>
<DirectoryMetadata>
        <FileSet>
                <FileName>GarciaCasella_001\.txt</FileName>
                <Description>
                        <Metadata name="eco.Titulo">Relaciones de la contabilidad social</Metadata>
                        ...
                        <Metadata name="eco.Autor">García Casella, Carlos Luís</Metadata>
                </Description>
        </FileSet>
        ...
        <FileSet>
                <FileName>PahlenAcuna-2004\.txt</FileName>
                <Description>
                        <Metadata mode="accumulate" name="eco.Titulo">Contabilidad social y ambiental</Metadata>
                        ...
                        <Metadata mode="accumulate" name="eco.Autor">Pahlen Acuña, Ricardo José María</Metadata>
                </Description>
        </FileSet>
</DirectoryMetadata>
```

Archivo .xsl de transformación

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="collection">
    <DirectoryMetadata>
        <xsl:apply-templates/>
    </DirectoryMetadata>
</xsl:template>
<xsl:template match="record">
    <FileSet>
    <FileName><xsl:value-of select="datafield[@tag='900']/subfield[@code='j']"/>\.txt</FileName>
        <Description>
            <xsl:apply-templates select="datafield[@tag='900']/subfield[@code='j']"/>
            <xsl:apply-templates select="datafield[@tag='903']/subfield[@code='a']"/>
            <xsl:apply-templates select="datafield[@tag='245']"/>
            <xsl:call-template name="Autores"/>
            ...
        </Description>
    </FileSet>
</xsl:template>
```

Archivo .xsl de transformación

```
        <xsl:template match="datafield[@tag='900']/subfield[@code='j']">
            <Metadata name="eco.NombreArchivo">
                <xsl:value-of select="."/>
            </Metadata>
        </xsl:template>
        <xsl:template match="datafield[@tag='903']/subfield[@code='a']">
            <Metadata name="eco.FechaCarga">
                <xsl:value-of select="."/>
            </Metadata>
        </xsl:template>
        <xsl:template match="datafield[@tag='245']">
            <Metadata name="eco.Titulo">
            <xsl:value-of select="subfield[@code='a']"/><xsl:if test="subfield[@code='b']">: <xsl:value-of select="subfield[@code='b']"
            /></xsl:if>
            </Metadata>
        </xsl:template>
        ...
        <xsl:template name="Autores">
            <xsl:for-each select="datafield[@tag='100']">
                <xsl:if test="not(subfield[@code='e'])"><xsl:value-of select="subfield[@code='a']"/>; </xsl:if>
            </xsl:for-each>
            <xsl:for-each select="datafield[@tag='700']">
                <xsl:if test="not(subfield[@code='e'])"><xsl:value-of select="subfield[@code='a']"/>; </xsl:if>
            </xsl:for-each>
        </xsl:template>
</xsl:stylesheet>
```

EJEMPLO BUSQUEDA EJEMPLAR

Buscar por autor o título

sql

MySQL Avanzado
Autor: SHUWARTZ, Baron
Ejemplares:2

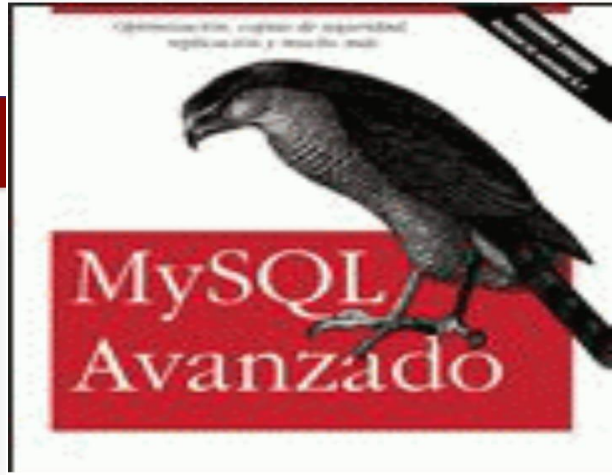
Microft sql server 2008 - Business Int...
Autor: VEERMAN, Erik
Ejemplares:3

SQL y JAVA. Guia para SWLJ, JDBC y...
Autor: MELTON, Jim
Ejemplares:1

SQL Server 2005, Designing a Databa...
Autor: MACKIN, J.C.
Ejemplares:1

Introducción al SQL para usuarios y p...
Autor: RIVERO CORNELIO, Enrique
Ejemplares:1

Optimizing and Maintaining a Databa...
Autor: THOMAS, Orin
Ejemplares:1



MYSQL AVANZADO

Autor	SHUWARTZ, Baron
Páginas:	832 p.
Biblioteca:	Biblioteca Central
Isbn:	978-84-415-2507-8
Estado	Disponible

Archivo .xsl de transformación

```
        <xsl:template match="datafield[@tag='900']/subfield[@code='j']">
            <Metadata name="eco.NombreArchivo">
                <xsl:value-of select="."/>
            </Metadata>
        </xsl:template>
        <xsl:template match="datafield[@tag='903']/subfield[@code='a']">
            <Metadata name="eco.FechaCarga">
                <xsl:value-of select="."/>
            </Metadata>
        </xsl:template>
        <xsl:template match="datafield[@tag='245']">
            <Metadata name="eco.Titulo">
            <xsl:value-of select="subfield[@code='a']"/><xsl:if test="subfield[@code='b']">: <xsl:value-of select="subfield[@code='b']"
            /></xsl:if>
            </Metadata>
        </xsl:template>
        ...
        <xsl:template name="Autores">
            <xsl:for-each select="datafield[@tag='100']">
                <xsl:if test="not(subfield[@code='e'])"><xsl:value-of select="subfield[@code='a']"/>; </xsl:if>
            </xsl:for-each>
            <xsl:for-each select="datafield[@tag='700']">
                <xsl:if test="not(subfield[@code='e'])"><xsl:value-of select="subfield[@code='a']"/>; </xsl:if>
            </xsl:for-each>
        </xsl:template>
</xsl:stylesheet>
```

Protocolos de interoperabilidad implementados por KOHA

Catalogación por copia. Protocolo Z39.50

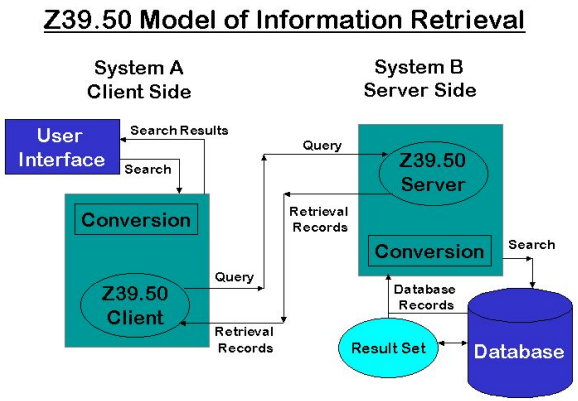
Catalogación en Koha

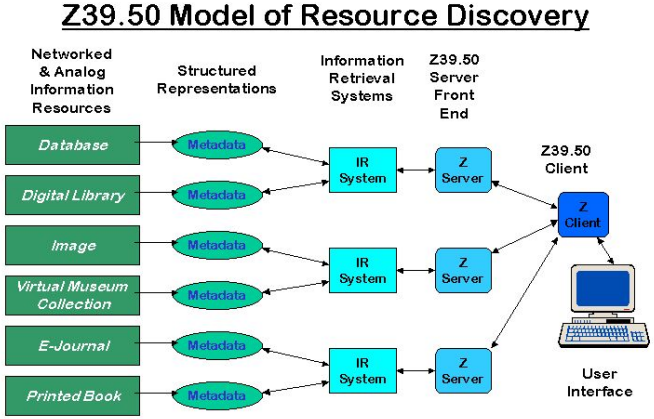
Catalogación por copia (Protocolo Z39.50)

¿Qué es el protocolo Z39.50?

Z39.50 es un protocolo cliente-servidor (preweb y de capa 7) dirigido a facilitar la búsqueda y recuperación de información en distintos sistemas a través de una misma interfaz. Su aplicación en el mundo de las bibliotecas y de los centros de documentación permite la consulta de recursos distribuidos en distintas bases de datos, desde un mismo punto de acceso.

Está cubierto por el estándar ANSI/NISO Z39.50 y el estándar ISO 23950.





Fuente: https://es.wikipedia.org/wiki/Z39.50 Gráficos: http://www.txmike.com/presentations/z3950/

Catalogación en Koha

Catalogación por copia con protocolo Z39.50)

Ingresando al panel de Administración tenemos acceso a la administración de servidores Z39.50

Parámetros adicionales

Servidores Z39.50/SRU

Definir los servidores externos para consultar datos MARC.

+ Nuevo servidor Z39.50 | + Nuevo servidor SRU

Administración de servidores Z39.50/SRU

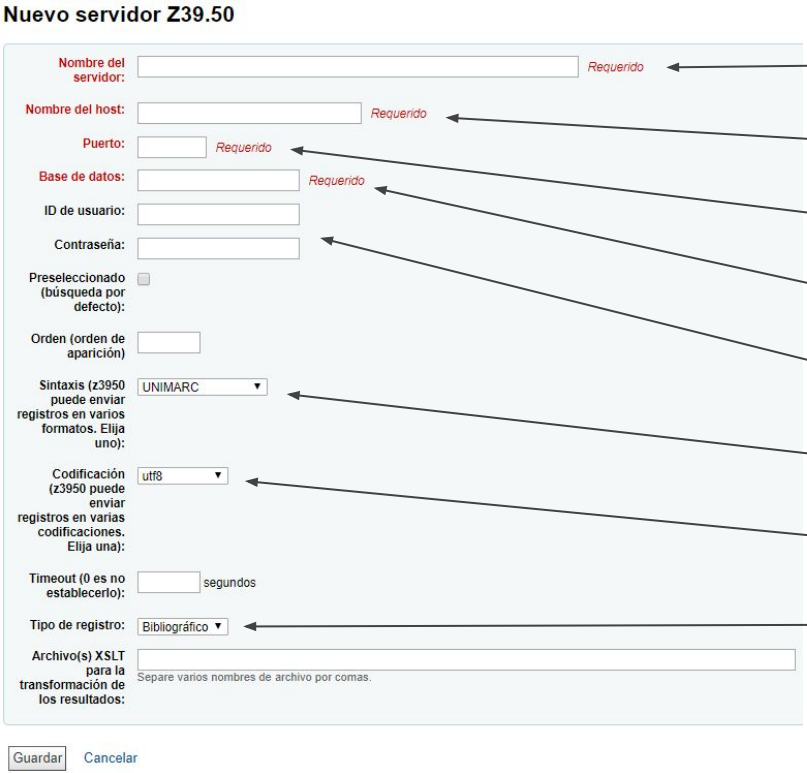
Mostrando 1 a 4 de 4 | Mostrar 20 entradas | Primero Previo Siguiente Último | Buscar:

Objetivo	Hostname/puerto	Base de datos	ID de usuario	Contraseña	Preseleccionado	Orden	Sintaxis	Codificación	Timeout	Tipo de registro	
LIBRARY OF CONGRESS	lx2.loc.gov:210	LCDB			Si	1	USMARC	utf8	0	Bibliográfico	Acciones
LIBRARY OF CONGRESS NAMES	lx2.loc.gov:210	NAF			Si	1	USMARC	utf8	0	Autoridad	Acciones
LIBRARY OF CONGRESS SRU	lx2.loc.gov:210	LCDB			No		USMARC	utf8	0	Bibliográfico	Acciones
LIBRARY OF CONGRESS SUBJECTS	lx2.loc.gov:210	SAF			Si	2	USMARC	utf8	0	Autoridad	Acciones

Mostrando 1 a 4 de 4 | Primero Previo Siguiente Último

Catalogación en Koha

Catalogación por copia (Protocolo Z39.50)



Nombre de identificación del servidor Z39.50

URL o IP del servidor

Puerto del servicio Z39.50 donde se realizan las consultas

Nombre de la Base de Datos

Existen servicios que requieren usuario y contraseña

La sintaxis expuesta por el servidor debe coincidir con la utilizada en la biblioteca cliente (Ej. Marc21)

Codificación expuesta por el servidor

El tipo de registro depende el servicio es para registros bibliográficos o registros de autoridades

Toda esta información es aportada por la entidad de la que pretendemos realizar catalogación por copia

Catalogación en Koha

Catalogación por copia (Protocolo Z39.50)

Ejercicio

1. Buscar en la WEB “argentina biblioteca nacional z39.50”.
2. Encontrar la información de Z39.50 requerida para incorporar el servidor .
3. Agregar el nuevo servidor.
4. Para verificar que funciona, ir al módulo de Catalogación, hacer click sobre el botón “**Nuevo desde Z39.50/SRU**”.
5. En la ventana emergente que se muestra, tildar uno o más servidores donde se desea realizar la búsqueda.
6. En el campo “Título”, escribimos “Juana Manso”.
7. Importar el registro seleccionado
8. Completar los campos requeridos según nuestra hoja de trabajo. Guardar y agregar un Item.

Catalogación en Koha

Catalogación por copia (Protocolo Z39.50)

Protocolo Z39.50

La Biblioteca Nacional Mariano Moreno brinda acceso a sus registros bibliográficos en formato MARC-21 a través del protocolo Z39.50. A fin de acceder al servicio puede ser que necesite efectuar en su unidad de información algunos ajustes básicos. Asegúrese de que su cliente utiliza los siguientes datos de configuración:

IP: 190.136.33.233
Puerto: 9991
Base: BNA01
Autentificación: SI
Usuario: Z39.50
Clave: Z39.50
Conjunto de caracteres: UTF-8
Sintaxis del Formato: MARC21

Autoridades

IP: 190.136.33.233
Puerto: 9991
Base: BNA10
Autentificación: SI
Usuario: Z39.50
Clave: Z39.50
Conjunto de caracteres: UTF-8
Sintaxis del Formato: MARC21

1-2 Buscamos en la WEB los parámetros para agregar como un nuevo servidor Z39.50 a la **Biblioteca Nacional "Mariano Moreno"**.

Resolución

Nombre del servidor: BIBLIOTECA NACIONAL "MARIANO MORENO" (AUT) *Requ*

Nombre del host: 190.136.33.233 *Requerido*
Incluye la parte del dominio, pero la parte de la ruta de la URL debe ir en la Base de datos.

Puerto: 9991 *Requerido*

Base de datos: BNA10 *Requerido*

ID de usuario: Z39.50

Contraseña: Z39.50

Preseleccionado (búsqueda por defecto):

Orden (orden de aparición)

Sintaxis (z3950 puede enviar registros en varios formatos. Elija uno): MARC21/USMARC

Codificación (z3950 puede enviar registros en varias codificaciones. Elija una): utf8

Timeout (0 es no establecerlo): segundos

Tipo de registro: Autoridad

Opciones adicionales SRU:
Opciones separadas por comas. Ejemplo: sru=get,sru_version=1.1. Vea además http://www.indexdata.com/yaz/doc/zoom.html.

Mapeo de campos de búsqueda SRU: Modificar

Archivo(s) XSLT para la transformación de los resultados:
Separe varios nombres de archivo por comas.

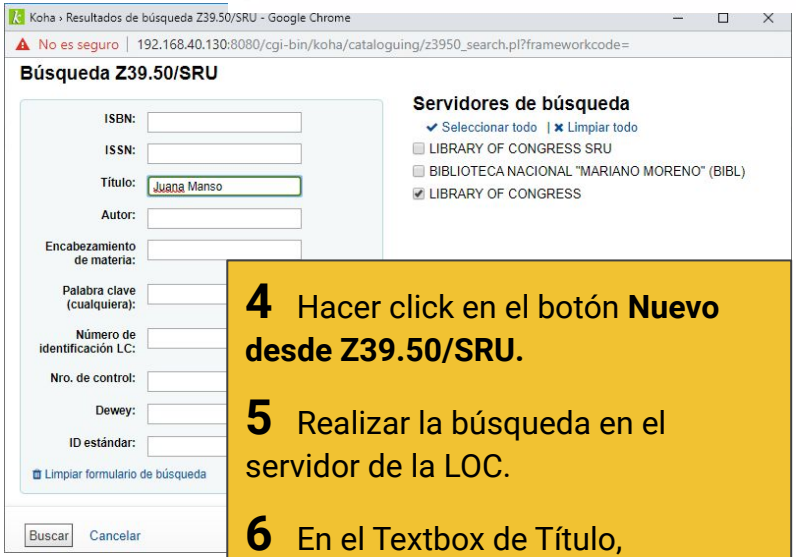
Guardar Cancelar

3 Agregamos los datos solicitados.

Catalogación en Koha

Catalogación por copia (Protocolo Z39.50)

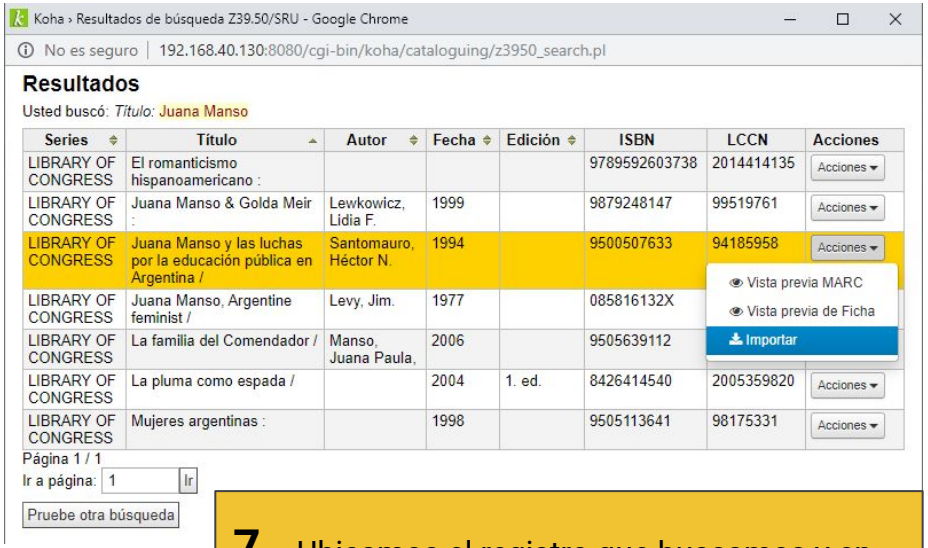
Resolución



4 Hacer click en el botón **Nuevo desde Z39.50/SRU.**

5 Realizar la búsqueda en el servidor de la LOC.

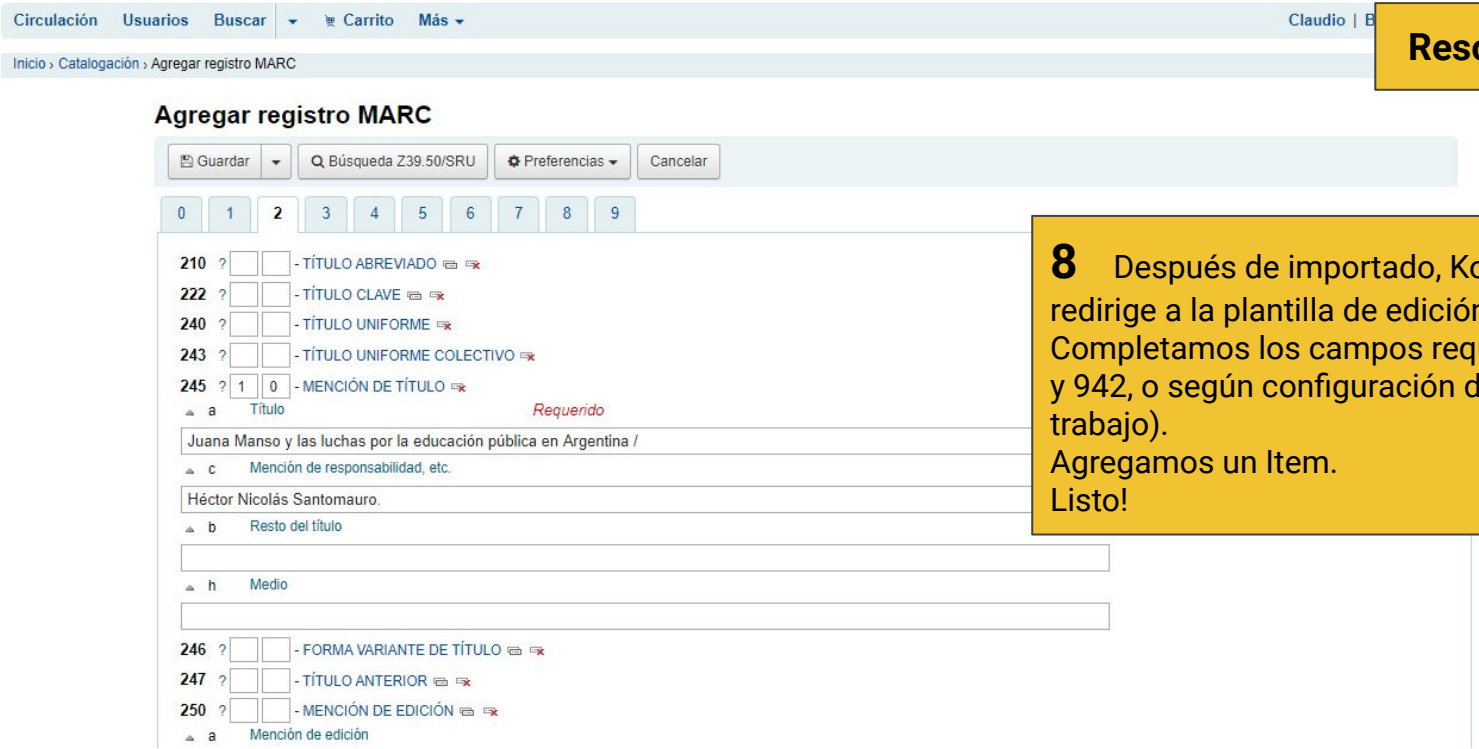
6 En el Textbox de Título, buscamos el término "**Juana Manso**" que forma parte del título.



7 Ubicamos el registro que buscamos y en **Acciones**, seleccionamos **Importar**.

Catalogación en Koha

Catalogación por copia (Protocolo Z39.50)



Resolución

8 Después de importado, Koha nos redirige a la plantilla de edición Marc. Completamos los campos requeridos (003 y 942, o según configuración de hoja de trabajo).
Agregamos un Item.
Listo!

z39.50, SRU y SRW

- **SRU** y **SRW**: equivalentes
  - **SRU**: API REST
  - **SRW**: API SOAP

En todos los casos se deben abrir puertos TCP en el firewall, salientes para buscar, entrantes para ofrecer.

Protocolos de interoperabilidad implementados por KOHA

- **REST**: Nueva (RESTful), vieja (svc), reportes.
- **Usuarios y circulación:**
  - SIP2
  - ILS-DI
- Otros: E-Payment, PayPal
- **Datos bibliográficos**
  - Z39.50 (servidor y cliente)
  - SRU/SRW (servidor y cliente)
  - OAI-PMH (servidor/dataProvider, cliente/serviceProvider aún no implementado)
  - Otros: unAPI, OpenSearch, RSS
- **Authentication**
  - LDAP
  - CAS
  - SAML
  - OAuth2

Protocolos implementados por KOHA

Protocolo OAI-PMH para exposición de los registros bibliográficos

```
pablo ~ $ curl -sv "http://www.uba.ar/"
*   Trying 157.92.5.125...
* TCP_NODELAY set
* Connected to www.uba.ar (157.92.5.125) port 80 (#0)
> GET / HTTP/1.1
> Host: www.uba.ar
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sat, 09 Jun 2018 07:08:23 GMT
< Server: Apache
< Vary: Accept-Encoding
< Content-Type: text/html
< Set-Cookie: ci_session=a%3A5%3A%7Bs%3A10%3A%22session_id%222a8a; expires=Sat, 09-Jun-2018 09:08:23 GMT;
path=/
< Transfer-Encoding: chunked
<
{ [842 bytes data]
<!DOCTYPE html>
<html lang="en"><head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<meta charset="UTF-8">
<title>Universidad de Buenos Aires</title>
<meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no"
name="viewport"><meta http-equiv="X-UA-Compatible" content="IE=Edge" />
<link
href='http://fonts.googleapis.com/css?family=Lora:400italic,700italic,700,400&amp;subset=latin,latin-ext'
rel='stylesheet' type='text/css'>
```

200 OK

HTML, porque es una página normal

```
pablo ~ $ curl -sv "http://catalogo.filo.uba.ar/cgi-bin/koha/oai.pl?verb=ListRecords&metadataPrefix=oai_dc"
*   Trying 157.92.88.40...
* TCP_NODELAY set
* Connected to catalogo.filo.uba.ar (157.92.88.40) port 80 (#0)
> GET /cgi-bin/koha/oai.pl?verb=ListRecords&metadataPrefix=oai_dc HTTP/1.1
> Host: catalogo.filo.uba.ar
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sat, 09 Jun 2018 07:17:20 GMT
< Server: Apache/2.4.10 (Debian)
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
< Content-Type: text/xml; charset=utf-8
<
{ [1031 bytes data]
<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type='text/xsl' href='/opac-tmpl/xslt/OAI.xslt'?><OAI-PMH
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd'
xmlns='http://www.openarchives.org/OAI/2.0/'><responseDate>2018-06-09T07:17:22Z</responseDate><request
metadataPrefix='oai_dc'
verb='ListRecords'>http://catalogo.filo.uba.ar/cgi-bin/koha/oai.pl</request><ListRecords><record><header
status='deleted'><identifier>KOHA-OAI-FILO:1</identifier><datestamp>Z</datestamp></header></record><record>
<header
status='deleted'><identifier>KOHA-OAI-FILO:2</identifier><datestamp>Z</datestamp></header></record><record>
<header status='deleted'><identifier>KOHA-OAI-FILO:3</identifier><datestamp>Z</
```

200 OK

XML, porque es un webservice

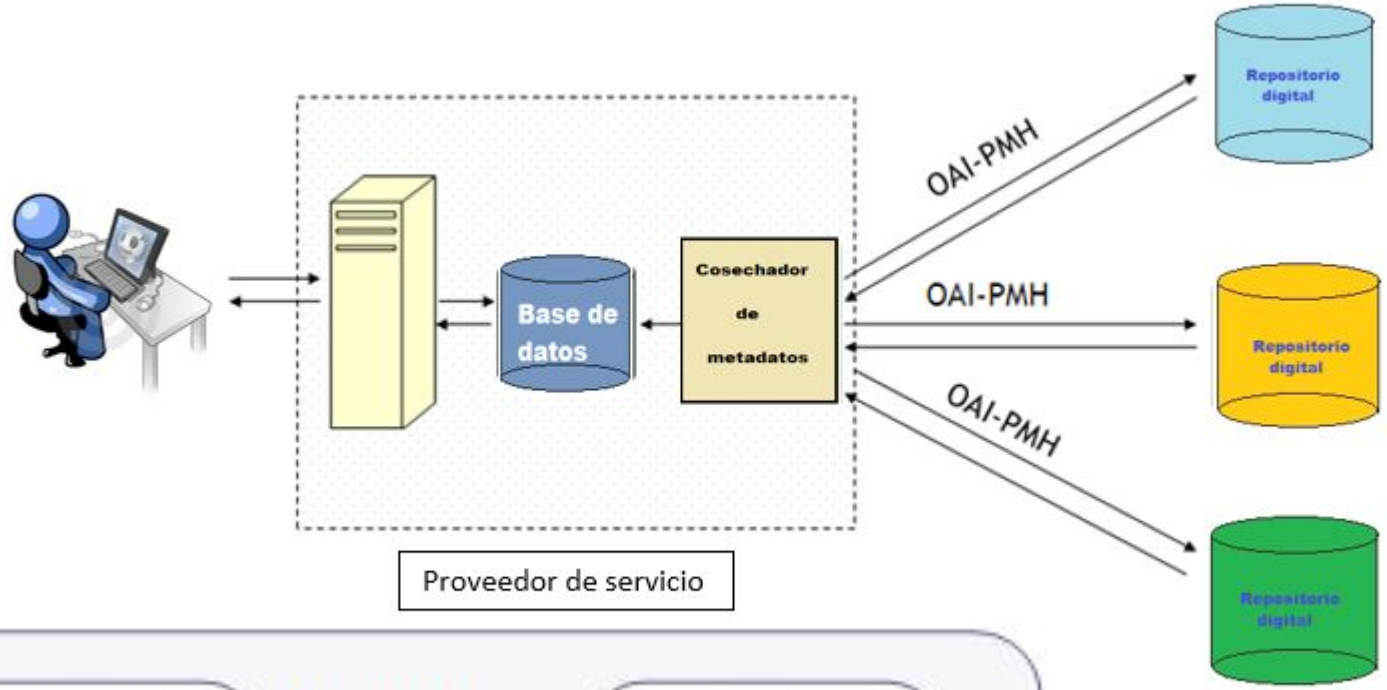
URI

Sintaxis (RFC 3986)

```
https://repodig.edu:8042/papers/oai.php?verb=Identify
\___/   \_______________/\_______________/ \____________/
  |             |                |              |
esquema     autoridad          ruta         consulta
        (dominio y puerto)
```

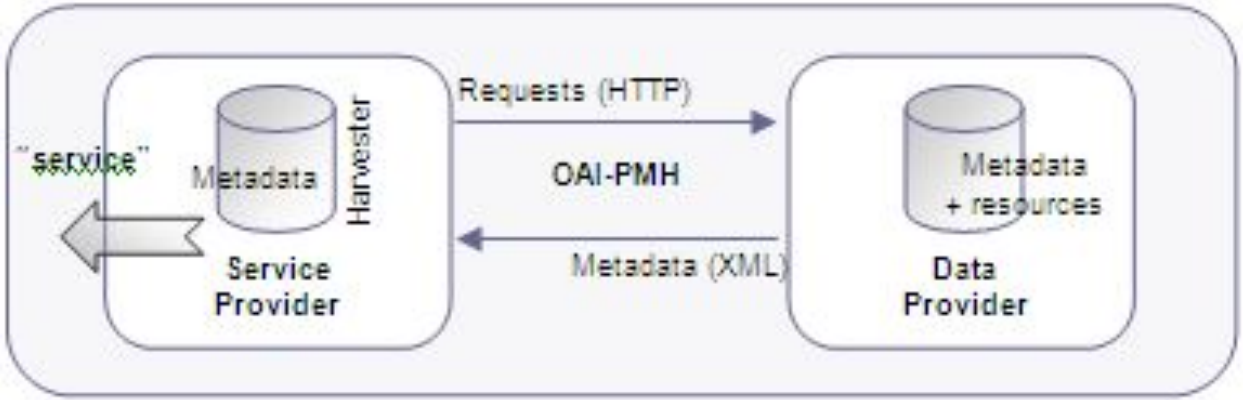
Dublin Core

- **Dublin Core** es un esquema de metadatos.
- DCMES v1.1: Dublin Core Metadata Element Set ("DC pelado")
  - **15** elementos de metadato: `Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage y Rights.`
  - Todos opcionales y repetibles.
- **DCMI Metadata Terms**: Vocabulario de un total de **55** elementos de metadato.



Proveedor de servicio

Proveedores de datos



Open **A**rchives **I**nitiative **P**rotocol for **M**etadata **H**arvesting

- OAI-PMH hace más sencilla la diseminación de la información.
- Es un protocolo que usa otro tipo de protocolo (http) para intercambiar archivos XML (de texto plano).
- Puede exponer los metadatos de objetos digitales o físicos.
- Su alcance cubre lo referido a los metadatos, no los recursos en sí.
- *No importa* el software mientras se respete el protocolo.
- ¿Por qué? Esencialmente **visibilidad** e **integración**, ¡y mucho más!
- ¿Qué? Publicaciones científicas, libros, revistas, ¡lo que sea!

Conceptos básicos

- **Cosechador (Harvester)**
  - Cliente OAI-PMH, software que realiza pedidos (requests) a repositorios.
  - Operado por un *service provider.*
- **Repositorio**
  - Colección de registros (con metadatos).
- **Recurso**
  - Objeto físico o digital a describir y exponer.
  - No se contempla cuando los metadatos están embebidos en el objeto digital.

Conceptos básicos

- **Item**
  - Agrupa todos los registros sobre un mismo recurso.
  - Puede no estar en un set, en uno o varios simultáneamente.
- **Registro (de metadatos)**
  - Conjunto de declaraciones/afirmaciones en **un** esquema de metadatos sobre **un** recurso. Pero se pueden tener **otros** registros. para describir un recurso en otros esquemas.
- **Identificador único**
  - URI que identifica de forma unívoca un recurso.
  - **Dos identificadores distintos**: Uno para el ítem y otro para el recurso digital.
- **Set/colección**
  - Conjunto predefinido de registros.

Actores

- **Data provider** (data: de *datum*, *dare*, dar; como donar y perdon)
  - **Expone** registros de metadatos de recursos, "*cual OPAC para computadoras*".
  - Es el **servidor OAI-PMH** (responde pedidos).
  - Koha, Omeka, CollectiveAccess, Samvera, DSpace
- **Service provider**
  - **Cosecha** registros. Además usualmente los guarda en una base de datos y los ofrece en un portal de búsqueda.
  - Contiene el **cliente OAI-PMH** (realiza pedidos)
  - Puede cosechar de distintas formas:
    - Cosechado total
    - En **un** dado conjunto/set/colección.
    - Entre determinadas fechas
  - Blacklight, vuFind, PKP
  - **Casos**
    - **DPLA**, Digital Public Library of America
      - MAP, Metadata Application Profile
    - **Europeana**
      - EDM, Europeana Data Model
    - **SNRD**

Se puede buscar en **un** set y entre fechas (o sólo desde/hasta).

No permite buscar la unión o intersección de conjuntos.

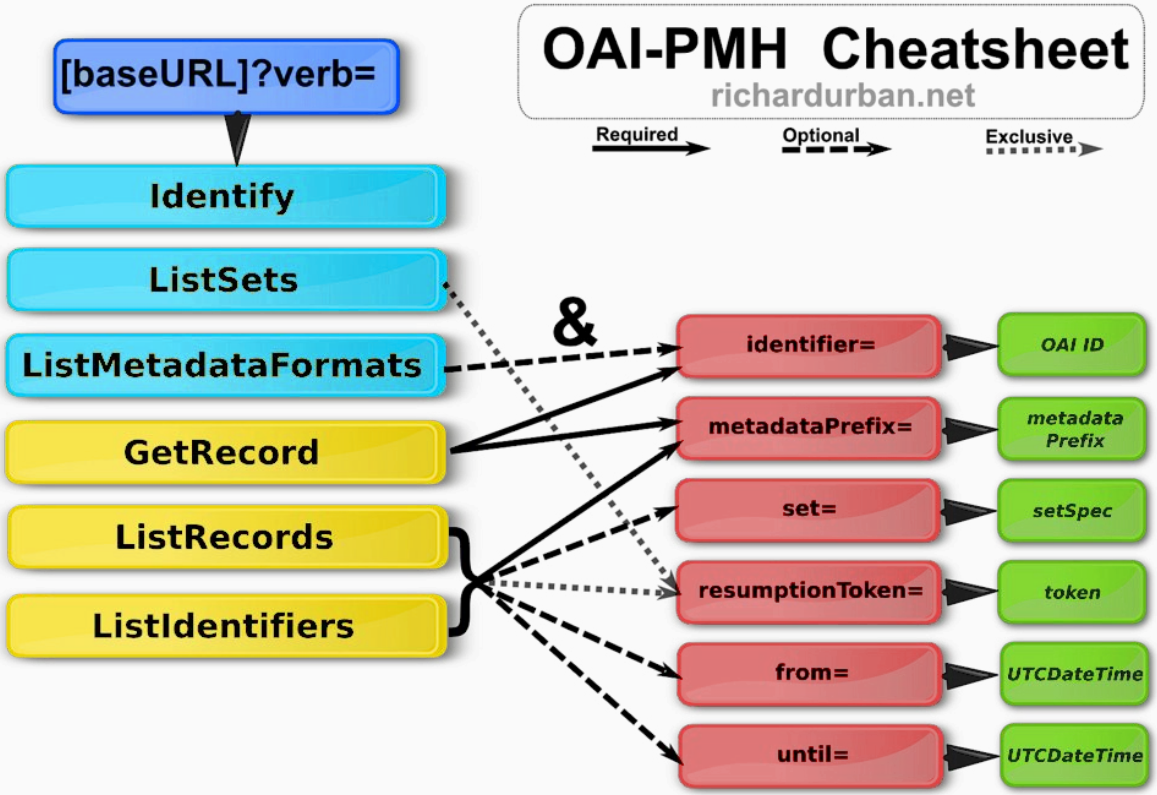
Europeana y DPLA tienen sus propios esquemas de metadatos basados en DC. Ambos (y muchos otros) son esquemas de metadatos distintos pero ambos se basan en DC y RDF. Comparten su adopción de OAI-PMH.

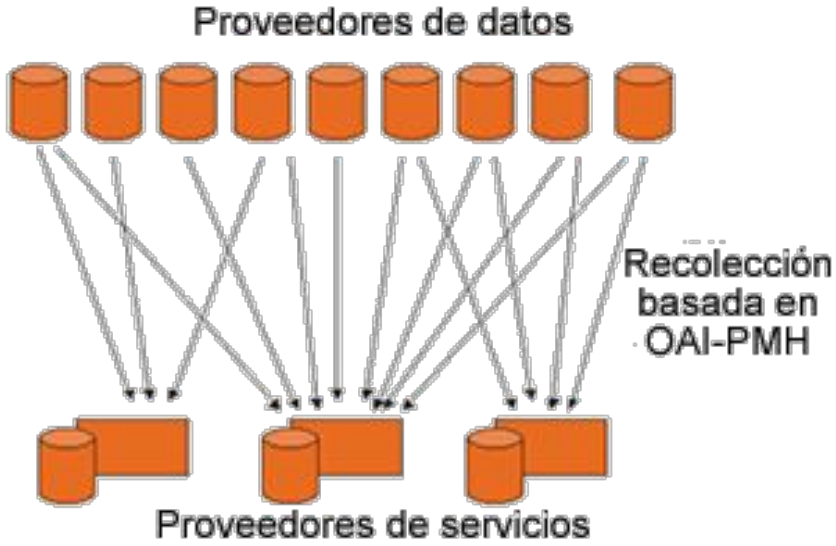
verbos

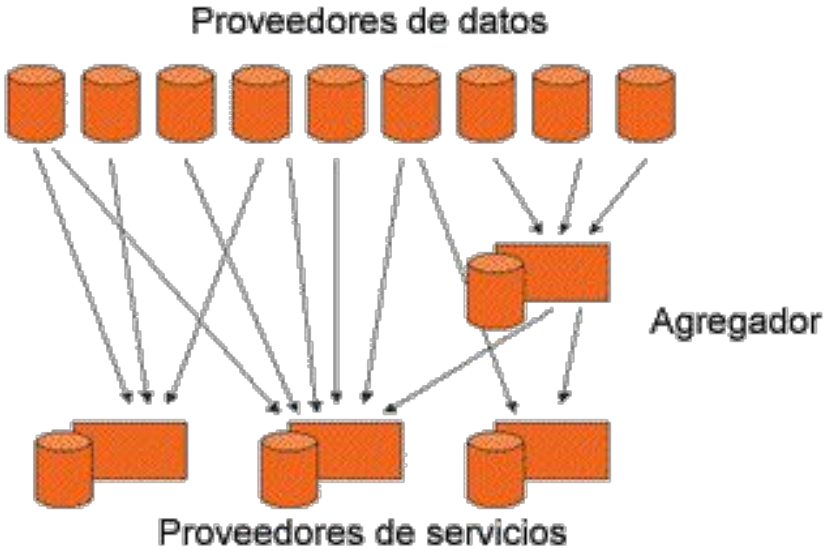
verbo	función
`Identify`	Describe el repositorio
`ListSets`	Lista conjuntos/sets disponibles
`ListMetadataFormats`	Lista esquemas de metadatos disponibles
`GetRecord`	Obtiene un registro dado un identificador*
`ListRecords`	Todos los registros [dado un conjunto rango de fechas]*
`ListIdentifiers`	Todos los identificadores [dado un conjunto o rango de fechas]*

* ¡Y dado un esquema de metadatos!

verbos


OAI-PMH Cheatsheet
richardurban.net
Required
Optional
Exclusive
[baseURL]?verb=
Identify
ListSets
ListMetadataFormats
GetRecord
ListRecords
ListIdentifiers
&
identifier=
metadataPrefix=
set=
resumptionToken=
from=
until=
OAI ID
metadata Prefix
setSpec
token
UTCDateTime
UTCDateTime





Imágenes tomadas del sitio del Min Cultura Español: http://travesia.mcu.es/portalnb/jspui/html/10421/1823/page3.htm

BDU2 como proveedor de servicio para los RI

SIUBDU2 REPOSITORIOS DIGITALES

Inicio | Explorar BDU2 | Estadísticas e indicadores | Otros Catálogos | Lenguaje

Buscá en 50 repositorios y bibliotecas digitales nacionales

Aqui la expresión a buscar | Todos los Campos | Buscar | Avanzado | Ayuda BDU2

Repositorios y Bib. Digitales Cosechados

Nombre del Repositorio	Cantidad de registros	Última fecha de colecta exitosa
ARGOS - Facultad de Humanidades y Cs Sociales - (UNMisiones)	648	2019-09-12 03:50:20
Biblioteca Digital (CDI - Ministerio de Hacienda de la Nación)	581	2019-09-12 04:03:54
Biblioteca Digital (UCA)	6854	2019-09-12 03:36:54

Sobre qué trata el material de BDU2?

cosechador.siu.edu.ar/bdu3/

Sistema Nacional de Repositorios Digitales

Pulsa F11 para salir del modo de pantalla completa

La producción científica argentina al alcance de todos

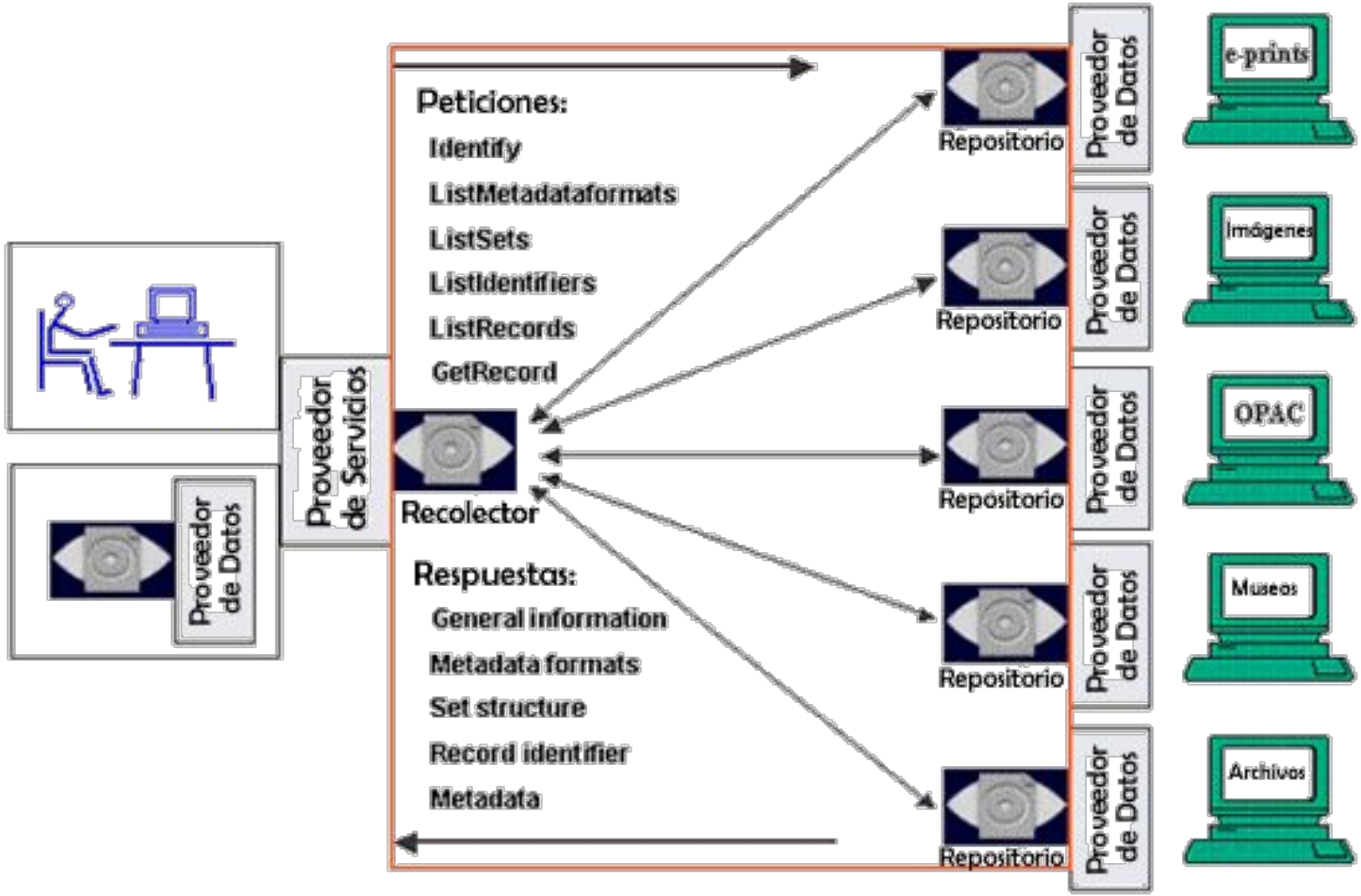
¿Qué estás buscando?

Todos los Campos

BÚSQUEDA AVANZADA

Portal de datos del SNRD. Ejemplo de agregador para los RI

235.481	34	34
publicaciones	repositorios	instituciones


Proveedor de Datos
Proveedor de Servicios
Peticiones:
Identify
ListMetadataformats
ListSets
ListIdentifiers
ListRecords
GetRecord
Recolector
Respuestas:
General information
Metadata formats
Set structure
Record identifier
Metadata
Repositorio
Repositorio
Repositorio
Repositorio
Repositorio
Proveedor de Datos
Proveedor de Datos
Proveedor de Datos
Proveedor de Datos
Proveedor de Datos
e-prints
Imágenes
OPAC
Museos
Archivos

Koha como OAI-PMH Data-Provider

Usted buscó: OAI

preferencias

▾ OAI-PMH

Preferencia	Valor
OAI-PMH	Habilite ▾ el servidor OAI-PMH de Koha.
OAI-PMH:archiveID	Identificar registros en este sitio con el prefijo KOHA-OAI-SIU . Por favor, ingrese el prefijo sin el carácter de finalización de dos puntos (":"). El arhiveID debe respetar la especificación OAI. Vea Directrices de implementación. Ejemplo: "oai:example-library.org"
OAI-PMH:AutoUpdateSets	Habilitar ▾ la actualización automática de conjuntos OAI-PMH cuando se crea o actualiza un registro bibliográfico
OAI-PMH:ConfFile	Archivo de configuración Servidor YAML Koha OAI: Si está vacío, el Servidor Koha OAI funciona en modo normal, de lo contrario, opera en modo extendido. En modo extendido, es posible que se pase como parámetro, otros formatos distintos a MARCXML o Dublin Core. OAI-PMH: ConfFile especifica a YAML un archivo de configuración, el cual, enumera los formatos de metadatados disponibles y archivos XSL utilizados para crearlos a partir de registros marcxml.
OAI-PMH:DeletedRecord	La tabla de Koha deletedbiblio nunca será vaciada o truncada (persistente) ▾ .
OAI-PMH:MaxCount	Solo retornar 100 registros a la vez en respuesta a una consulta ListRecords o ListIdentifiers.

Guarde todas las preferencias Cancelar

En primer lugar es necesario habilitar el protocolo, indicar un prefijo para la identificación de los registros e indicar la cantidad de registros por página que resultarán expuestos.

CIN-SIU-BDU rol Service provider

Una vez habilitado el protocolo OAI-PMH, es posible reportar la URL a la BDU para que resulte cosechado, con lo que se realizan dos actualizaciones regulares al año del catálogo y no se envían más registros en forma manual al catálogo cooperativo.

No es seguro | bdu.siu.edu.ar/prod/bibliotecas/distrocoop.php

Qué instituciones cooperan con la BDU? Clasificados por formato bibliográfico reci

63 Cooperantes en Formato MARC21

Acrónimo	Institución	Dependencia	Actualización	Registros	Método
BDUM	Base de Datos MARC21 - OCLC	BDUMARC21	07/07/2007	158030	👤
bnm	Biblioteca Nacional de Maestros	BNM	12/12/2018	174125	👤
cnea	Comision Nacional de Energia Atomica (CNEA)	Sistema de Bibliotecas	07/05/2019	13578	↻
cin	Consejo Interuniversitario Nacional	Biblioteca Digital	05/03/2018	982	↻
cpce	Consejo Prof. Cs.Económicas CABA	Centro de Inform. Bibliográfica	09/03/2018	9757	👤
flacso	FLACSO	Biblioteca de	21/06/2019	36881	↻

↻ Cooperante cosechado automáticamente por OAI-PMH
👤 Cooperante que realiza envío de bases de datos

No es seguro | 192.168.56.103/cgi-bin/koha/oai.pl?verb=Identify

Koha OAI-PMH Data Provider

Identify | Sets | Records | Identifiers | Metadata Formats

Response Date 2019-09-26 15:49:03

Repository Information

Repository Name	
E-Mail Contact	root@localhost
Repository identifier	
Sample identifier	
Protocol Version	2.0
Earliest Registered Date	2019-09-20 23:33:03
Date Granularity	YYYY-MM-DD hh:mm:ss
Deletion Mode	persistent

Powered by Koha

Ejercitación básica OAI-PMH

Ejercitación

- Habilitar el protocolo OAI-PMH, establecer el prefijo adecuado para las bibliotecas y establecer la paginación en 100 registros.
- Verificar la URL de KOHA en el proveedor de servicio OAI-PMH. Verificar el verbo identiy y encontrar en la documentación de KOHA, la vía de poder alterar los datos que aparecen con el verbo Identify.
- Invocar el verbo ListMetadataFormats. Proponer una explicación para los diferentes schemas de metadatos que son expuestos a través del protocolo.
- Usando el verbo ListRecords, detallar las diferencias entre los diferentes formatos que vienen out-of-the box en Koha y determinar cuál de ellos es el apropiado para resultar cosechado por BDU.

Ejercitación

- ¿Puedo buscar por autor/título/*campo* con OAI-PMH?
- ¿Puedo buscar *pintores franceses del sXVIII con al menos 5 obras*?
- Tome una de las siguientes URL-base de repositorios OAI-PMH:
  - http://biblio.econ.uba.ar/cgi-bin/koha/oai.pl (Koha)
  - http://catalogo.filo.uba.ar/cgi-bin/koha/oai.pl (Koha)
  - http://digital.bl.fcen.uba.ar/gsdl-282/cgi-bin/oaiserver.cgi (Greenston 2.82)
  - http://ri.conicet.gov.ar/oai/request (DSpace)
  - http://www.caicyt-conicet.gov.ar/dila/oai-pmh-repository/request (Omeka)
  - http://revistascientificas.filo.uba.ar/index.php/afc/oai
  - http://oai.europeana.eu/oaicat/OAIHandler (OAICat, Jeff Young, OCLC)

- Obtenga
  - Información del repositorio ([URL-base]?verb=Identify)
  - Liste sets disponibles ([URL-base]?verb=ListSets)
  - Liste registros ([URL-base]?verb=ListRecords&metadataPrefix=oai_dc)
- Realice lo mismo pero utilizando
  - **OAI - Repository Explorer** http://re.cs.uct.ac.za/
  - **OAI-PMH Validator** http://validator.oaipmh.com/

- Configure su Koha para permitir ser cosechado
  - `OAI-PMH` → Habilitar
  - `OAI-PMH:archiveID` (no haga chanchadas, normalice con URN)
    - http://www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm
    - oai-identifier = scheme ":" namespace-identifier ":" local-identifier
    - Ejemplo: `oai:miinstitucion.org:biblio`
      - `oai:miinstitucion.org:biblio:<biblionumber>`
  - `OAI-PMH:AutoUpdateSets`
- Si dispone de registros
  - Cree un conjunto
    - **setSpec** (no haga chanchadas, use una URN). Ejemplos:
      - setSpec: tesis:doctorado, fotos, publicaciones:revistaHola
      - setName: “Tesis de doctorado”, “Fotos”, “Revista Hola”
  - Establezca el mapeo (correspondencia) con el criterio por el cual algunos registros formarán parte del conjunto.
  - ¡Coséchese!

OAI-PMH @ Koha

- **XSLT**
  - **XML → XML**
  - **Mapeo**

MARC	Dublin Core
100, 110, 111, 700, 710, 711	`contributor`
008/7-10, 260 $c $g	`date`
020$a, 856 $u	`identifier`
008/35-37, 041, 546	`language`
260 $a $b	`publisher`
050, 060, 082, 600, 610, 611, 650	`subject`
245, 246	`title`
Leader 06, Leader 07	`type`

Schemas de metadatos posibles de cosechar

List of Metadata Formats

Results fetched 0

oai_dc LIST RECORDS

Namespace http://www.openarchives.org/OAI/2.0/oai_dc/

Schema http://www.openarchives.org/OAI/2.0/oai_dc.xsd

marc21 LIST RECORDS

Namespace http://www.loc.gov/MARC21/slim http://www.loc.gov/standards/marcxml/schema/MARC21slim

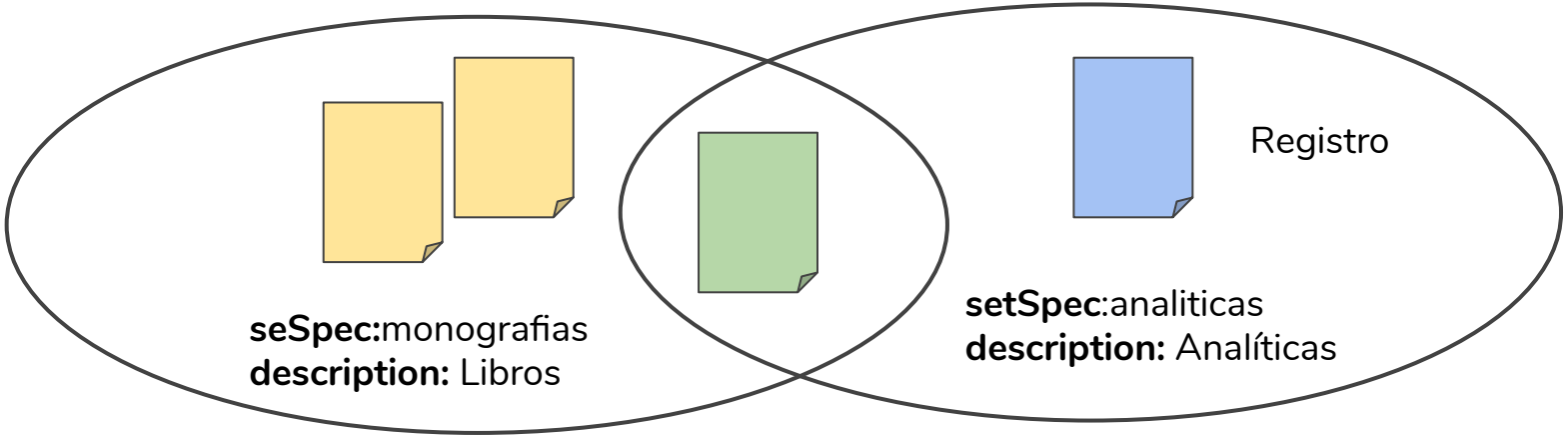
Schema http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd

marcxml LIST RECORDS

Namespace http://www.loc.gov/MARC21/slim http://www.loc.gov/standards/marcxml/schema/MARC21slim

Schema http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd

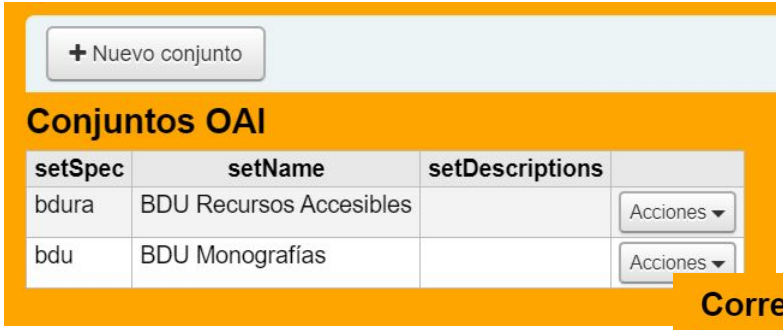
Protocolo OAI-PMH Sets



El protocolo OAI-PMH permite definir conjuntos entre los registros expuestos. Así un registro pertenecerá a 0,1 o más conjuntos. Estos conjuntos, permiten hacer lo que se denomina “Cosecha Selectiva”. El ejemplo más famoso de Cosecha Selectiva, lo hace el MINCyT con el set SNRD. Koha de igual manera, nos permite definir uno o más conjuntos de acuerdo a valores de los campos de los registros, los cuales serán usados para crear los sets.

Protocolo OAI-PMH Sets

A través de la opción de Administración que permite definir los conjuntos OAI entonces, tenemos que dividir las acciones en dos: Por un lado la creación de sets, y por el otro, la definición de correspondencia. La definición de correspondencia se refiere a condicionales sobre los campos de MARC21 que serán los que chequeará para incluir o no en un set en particular.



Es una limitación de (la interfaz) de Koha que la correspondencia solo tiene uniones lógicas a través del operador OR y no es posible definir condicionales basados en el Leader

Actualización de los sets

Luego de realizar la definición de los sets es necesario correr el script build_oai_sets.pl. Esta acción la podemos realizar con uno de los comandos de Koha incluídos en el paquete de Debian

```
koha-foreach --chdir --enabled /usr/share/koha/bin/migration_tools/build_oai_sets.pl
```

Esta acción recorre los registros y los incluye o no en los sets creados (retrospectivamente), con lo que luego de su ejecución podemos chequear si es que efectivamente con la invocación del verbo se ven o no incluídos en el set.

Para evitar correr el proceso en forma cronológica, es que se puede setear la preferencia

	arhiveID debe respetar la especificación OAI. Vea Directrices de implementación. Ejemplo: "oai:example-library.org"
OAI-PMH:AutoUpdateSets	Habilitar ▼ la actualización automática de conjuntos OAI-PMH cuando se crea o actualiza un registro bibliográfico
OAI-PMH:ConfFile	Archivo de configuración Servidor YAML Koha OAI: Si está vacío, el Servid

Verificación del funcionamiento de OAI-PMH (y los sets)

No es seguro | 192.168.56.103/cgi-bin/koha/oai.pl?verb=ListRecords&set=bdura&metadataPrefix=marcxml

Koha OAI-PMH Data Provider

Identify Sets Records Identifiers Metadata Forma

Response Date 2019-09-27 05:46:43

List of Records

Results fetched 2

192.168.56.103/cgi-bin/koha/oai.pl?verb=ListRecords&set=bdura&metadataPrefix=marcxml

Sets

Metadata

Identifier KOHA-OAI-SIU:2

Last Modified 2019-09-27 05:29:09

En la pantalla de presentación de los registros que se visualiza por OAI-PMH podemos ver por un lado la identificación del registro que el service provider potencialmente puede usar para detectar modificaciones (cosecha incremental). También podemos verificar los sets a los cuales pertenece cada registro.

192.168.0.30/cgi-bin/koha/oai.pl?verb=ListRecords&metadataPrefix=marcxml&set=rian

192.168.0.30/cgi-bin/koha/oai.pl?verb=ListRecords&metadataPrefix=oai_dc&set=rian

También podemos analizar los diferentes schemas de metadatos expuestos de acuerdo al schema solicitado en el verbo.

Definición de sets OAI-PMH

Ejercitación

- En la Jornada 1, se ha definido un nuevo tipo de ítem relacionado con los recursos accesibilizados por la Biblioteca. Los mismos deben ser expuestos en un set propio para resultar cosechados por SIU-BDU-RA.
- Realizar la implementación del set, de acuerdo a la especificación del tipo de ítem, y verificar que sólo estos registros resulten expuestos en ese set para cosecha.

Definición de sets OAI-PMH

Ejercitación Integrada

- La biblioteca de Postgrado comenzará a interoperar con el catálogo cooperativo de la red de bibliotecas del área de Educación a distancia. Dicha red cosechará usando el protocolo OAI-PMH sobre el KOHA central que la instiutución dispone. El setSpec debe llamarse “ariadne” en minúscula y debe exponer de a 100 registros por página. Cosechará en el schema de metadatos oai_dc.

  La red solo manifiesta interés en las publicaciones originales de edición EDIUNN, es decir que el campo 260$b incluya en su texto EDIUNN. Debe modelizarse el proceso considerando un KOHA central para toda la institución, la cual, obviamente en diferentes branches puede haber publicaciones de edición CNEA, sin embargo deben exponerse solo los títulos de la biblioteca de Postgrado.

Definición de sets OAI-PMH

Ejercitación Integrada

- Dadas las limitaciones de KOHA para el armado de sets, la Biblioteca de Postgrado se aviene a incorporar un flag que indicará de aquí en adelante en el campo 990$z, para indicar si es que el registro es que debe publicarse en dicho set o no, sin embargo solicita que la incorporación retrospectiva de dicho flag se realice en forma automática con algún tipo de procesamiento automatizado, dado que el campo Editor se halla correctamente cargado y en caso contrario, implicaría una revisión manual de más de 1600 registros bibliográficos.
- Analizar las diferentes opciones de cambios globales como para poder implementar este requerimiento.

MarcEditor: koha (1).mrk

File Edit Fonts Reports Tools OCLC WorldCat Plug-ins Help

What would you like to d

```
=LDR  01349cam a2200337la 4500
=001  59799
=003  OSt
=005  20190921190627.0
=008  081103s2009\\\\sp\a\\\\\\\\\001\0\spa\d
=020  \\$a9788441524675
=040  \\$bSPA$dAgFBB$cOSt
=041  \\$aSPA$hENG
=044  \\$cSP
=100  1\$aDeitel, Paul J.
=240  00$aAjax, Rich Internet Applications and W
=245  10$aAjax, Rich Internet Applications y desa
=260  \\$aMadrid$bCNEA$c[2009]
=300  \\$a1088 p.$bil.$c23 cm.$flmpreso
=500  \\$aIncluye índice.
=500  \\$aTraducción de: Ajax, Rich Internet Appli
=650  \4$aAJAX (Desarrollo de tecnología de sitio
=650  \4$aJavaScript (Lenguaje de programación
=650  \4$aDesarrollo de páginas Web.
=650  \4$aPáginas Web$xDiseño.
=650  \0$aPROGRAMACIÓN WEB$93
=700  1\$aDeitel, Harvey M,$d1945-
=700  1\$aGómez Celador, José Luis,
=942  \\$2ddc$cBK
=990  \\$brian
=990  ##$brian
=990  \\$aSistema de Bibliotecas de la Universida
=999  \\$c1$d1
=952  \\$00$10$2ddc$40$70$91$aCEN$bCEN$d2019-09-17$i93838$p93838$r2019-09-17$w2019-09-17$yBK

=LDR  01082pam a2200361 a 4500
=990  ##$brian
=999  \\$c2$d2
```

MarcEdit.NET Batch Editing Tools

Add/Delete Field Utility

- Add/Delete Field
- Copy Field Data
- Edit Fields
- Edit Indicators
- Edit Subfield Data
- Swap Fields

Field: 990

Field Data: \\$brian

Find What: //^=260.*\$bCNEA//

General Options
- [ ] Match case
- [x] Use Regular Expression
- [ ] Process batch operation

Add Field Options
- [ ] Insert before
- [ ] Insert last
- [ ] Add field only if not present
- [ ] Add field only if not a duplicate
- [x] Add Field If Present/NOT Present

Delete Field Options
- [ ] Remove Duplicate Data
- [ ] Remove by field position
- [ ] Remove if field data does n

Delete Field | Add Field | Close

regextester.com

RegEx Testing
From Dan's Tools

Web Dev | Conversion | Encode/De

Regular Expression (Javascript, flags)

/^=260.*\$bCNEA/g — 1 match

Test String

```
=260  \\$aMadrid$bCNEA$c[2009]
```

Substitution

Protocolos aceptados por KOHA

Protocolo SRU para registros bibliográficos

loc.gov/standards/sru/

The Library of Congress » Standards » SRU

SRU Web Pages ▾ SEARCH

SRU Search/Retrieval via URL

SRU (the protocol)
CQL (query language)
ZeeRex (service description)

SRU/CQL

SRU- Search/Retrieve via URL - is a standard XML-based protocol for search queries, utilizing **CQL - Contextual Query Language** - a standard syntax for representing queries.

SRU IS Simple! | Cool SRU Sites

Listserv

Specifications:

SRU 2.0 **SRU 1.2** **SRU 1.1** **CQL**

Version 1.1 was released in 2004 and 1.2 in 2007. SRU 2.0 is an OASIS Standard.

Companion Specs:

Scan **Explain** **SRW** **XML Files**

Also

- Conformance (SRU Base Profile)
- Transport
- Diagnostics
- Record Schemas
- Extensions
- Profiles

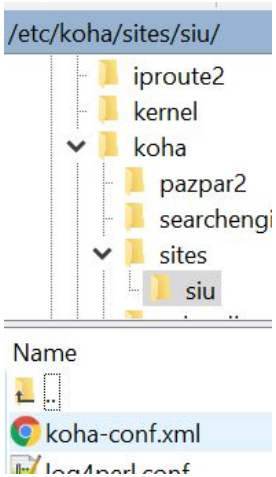
Resources

Especificación de SRU (Library Of Congress)

Protocolo SRU

El protocolo SRU (Search Retrieval vía URL) permite recuperar y realizar búsquedas a través de http, usando el indexador que esté actualmente configurado con Koha. En nuestro caso Zebra.

En los ejemplos que vamos a revisar, el protocolo nos estará devolviendo registros bibliográficos en formato MARC-XML, que es de esta manera es que se encuentran almacenados en Zebra. Podremos aplicar operadores booleanos y límites para las consultas que hemos visto con el protocolo Z39.50 pero usando URLs



Para habilitar el protocolo, tendremos que efectuar un conjunto de cambios en el archivo de configuración de la instancia.

Los cambios consisten en definir (descomentar) las áreas de configuración las que implementan el servidor público de SRU.

Todo este trabajo apunta a contar con un **servicio que permita buscar sobre el índice de Koha** y brinde la información de modo que sea procesable por otra componente de software.

Cambios a realizar para habilitar el servicio de SRU (publicserver)

```
<!-- Uncomment the following entry if you want to run the public Z39.50 server.
    Also uncomment the <server> and <serverinfo> sections for id 'publicserver'
    under PUBLICSERVER'S BIBLIOGRAPHIC RECORDS title-->

<listen id="publicserver" >tcp:@:9990</listen>
```

```
<!-- PUBLICSERVER'S BIBLIOGRAPHIC RECORDS -->
<!-- This can be used to set up a public Z39.50/SRU server.

<server id="publicserver"  listenref="publicserver">
    <directory>/var/lib/koha/siu/biblios</directory>
    <config>/etc/koha/sites/siu/zebra-biblios-dom.cfg</conf
    <cql2rpn>/etc/koha/zebradb/pqf.properties</cql2rpn>

  <xi:include href="/etc/koha/marc21-retrieval-info-bib-dom
              xmlns:xi="http://www.w3.org/2001/XInclude">
    <xi:fallback>
      <retrievalinfo>
       <retrieval syntax="marc21" name="F"/>
```

```
<xi:include href="/etc/koha/zebradb/explain-biblios.xml"
            xmlns:xi="http://www.w3.org/2001/XInclude">
  <xi:fallback>
  <explain xmlns="http://explain.z3950.org/dtd/2.0/">
      <serverInfo>
        <host>localhost</host>
        <port>9990</port>
        <database>biblios</database>
      </serverInfo>
    </explain>
  </xi:fallback>
</xi:include>
```

root@siukoha64: ~

```
root@siukoha64:~# koha-zebra --stop siu
root@siukoha64:~# koha-zebra --status siu
 * Zebra not running for siu                                    [fail]
root@siukoha64:~# koha-zebra --start siu
root@siukoha64:~# koha-zebra --status siu
 * Zebra running for siu                                        [ OK ]
root@siukoha64:~#
```

No es seguro | 192.168.56.103:9990/biblios?version=1.1&operation=searchRetrieve&query=ajax&maximumRecords=3

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<zs:searchRetrieveResponse xmlns:zs="http://www.loc.gov/zing/srw/">
  <zs:version>1.1</zs:version>
  <zs:numberOfRecords>1</zs:numberOfRecords>
 ▼<zs:records>
  ▶<zs:record>...</zs:record>
  </zs:records>
 ▼<zs:echoedSearchRetrieveRequest>
   <zs:version>1.1</zs:version>
   <zs:query>ajax</zs:query>
   <zs:maximumRecords>3</zs:maximumRecords>
   <zs:recordPacking>xml</zs:recordPacking>
  </zs:echoedSearchRetrieveRequest>
```


Qué obtenemos con el publicserver?

(en principio MARC-XML)

Ejercitación protocolo SRU

- El área de Comunicación de la Universidad está implementando un buscador para el sitio institucional, el mismo actuará sobre las noticias y los contenidos generales. La Biblioteca, solicita al área de IT que el catálogo en línea esté integrado a los resultados de dicho buscador. El nivel de integración mínimo es que al menos se indique la biblioteca posee material sobre el tema buscado y se oferte un link del estilo:

  La biblioteca posee X cantidad de publicaciones sobre el término buscado! Hacé click <aquí> para verlos.

  Implementar el servicio de búsqueda con Koha, especificar el link con la expresión de búsqueda y verificar básicamente que el número de resultados ofertados por el servicio son los mismos que los ofertados por el catálogo en línea. Simplificar al máximo el XML devuelto.

Protocolos aceptados por KOHA

Protocolo ILS-DI para interoperabilidad con vuFind

Fork me on GitHub

vuFind
Search. Discover. Share.

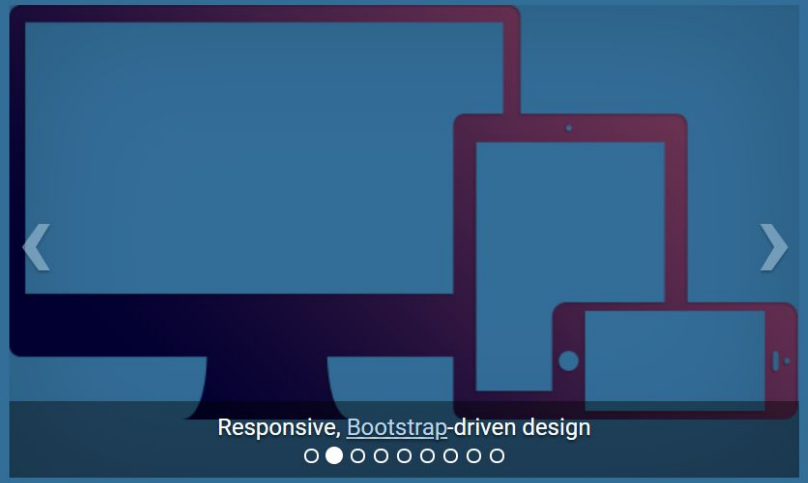
About Features Downloads Getting Started Documentation Events Support

VuFind is a library resource portal designed and developed for libraries by libraries. The goal of VuFind is to enable your users to search and browse through all of your library's resources by replacing the traditional OPAC to include:

- Catalog Records
- Locally Cached Journals
- Digital Library Items
- Institutional Repository
- Institutional Bibliography
- Other Library Collections and Resources

VuFind is completely modular so you can implement just the basic system, or all of the components. And since it's open source, you can modify the modules to best fit your need or you can add new modules to extend your resource offerings.

Live Demo



News

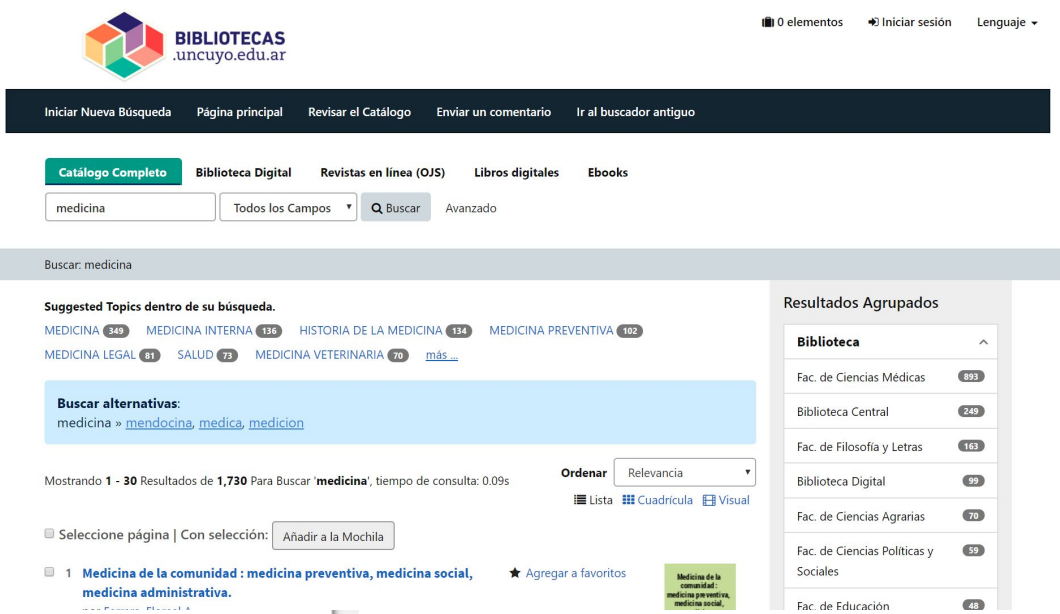
- Release 6.0.1 - 9/16/2019
- Release 6.0 - 7/15/2019

VILLANOVA UNIVERSITY

VuFind

Mostly written in PHP

Qué es vuFind?



Ejemplos de vuFind

University of California
CDL
California Digital Library

Staff Directory Contact CDL

Search

About CDL | Services and Projects | Resources For… | Committees and Groups

California Digital Library > CDLINFO > Digital Library Federation (DLF) releases Technical Recommendation from ILS Discovery Interface Task Group

Digital Library Federation (DLF) releases Technical Recommendation from ILS Discovery Interface Task Group

August 25, 2008 | Author: Ellen Meltzer | Categories: Newsletter, Technology

By Peter Brantley, Executive Director of the Digital Library Federation

Such standard interfaces will allow libraries to deploy new discovery services to meet ever-growing user expectations in the Web 2.0 era, take full advantage of advanced ILS data management and services, and encourage a strong, innovative community and marketplace in next-generation library management and discovery applications.

A group of eight professionals from major North American research libraries, including Patricia Martin, CDL Director of Bibliographic Services, prepared the recommendation during late 2007 and early 2008. Members of the group surveyed the library community about their needs, made presentations, and held open discussions face to face and online with librarians, developers, and vendors.

preferencias

▼ILS-DI

Preferencia	Valor
ILS-DI	Habilitar ▼ Servicios ILS-DI para usuarios del OPAC (disponible en: /cgi-bin/koha/ilsdi.pl)

Guarde todas las preferencias Cancelar

En primer lugar, es preciso habilitar el protocolo desde las preferencias del sistema, de modo de habilitar el acceso al script del protocolo.

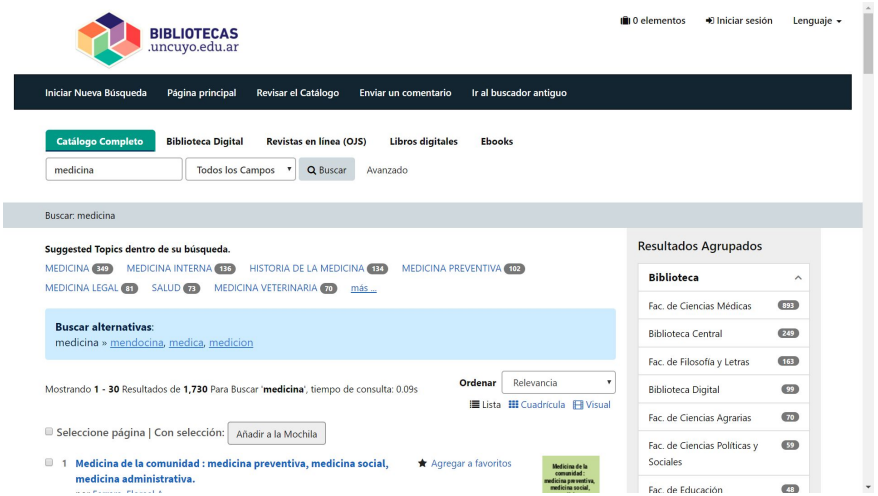
Es un protocolo generado en el marco de DLF (Digital Library Federation) y es abreviatura de

Internet Library System Discovery Interface

(ILS-DI).

El protocolo en el marco de KOHA se encuentra autodocumentado y permite ofrecer un conjunto de servicios a otras aplicaciones que quieran conectar y conocer información (primariamente pública).

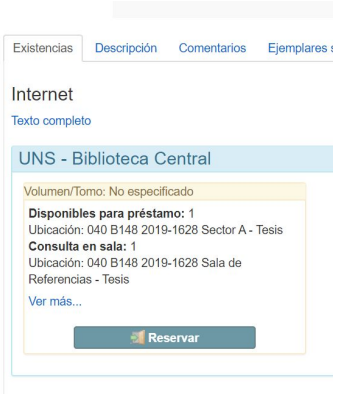
Para qué serviría el protocolo ILS-DI en el marco de la interoperabilidad vuFind & Koha?



OAI PMH Service provider

Desde vuFind cosechamos por OAI-PMH Koha

koha
FREE LIBRARY SYSTEM



Usando el identificador obtenido, podemos consumir el protocolo ILS-DI de Koha para presentar la información de disponibilidad y más servicios

Cuáles “más servicios”?
En verdad el protocolo está autodocumentado

No es seguro | 192.168.56.103/cgi-bin/koha/ilsdi.pl

koha | Carrito | Listas | Idiomas | Bienvenido, Emiliano Marmonti | Historial de búsqueda [x] | Salir

Buscar | Catálogo de bibliotec | Ir

Búsqueda avanzada | Búsqueda de autoridad | Nube de etiquetas

Inicio › ILS-DI

Important links here.
Important links here.

ILS-DI

Nivel 1: Interfaces de descubrimiento básicas

- HarvestBibliographicRecords *(Utilice OAI-PMH en su lugar)*
- HarvestExpandedRecords *(Utilice OAI-PMH en su lugar)*
- GetAvailability
- GoToBibliographicRequestPage *(Usar en cambio OPAC)*

Nivel 2: Suplemento de OPAC elemental

- HarvestAuthorityRecords *(Utilice OAI-PMH en su lugar)*
- HarvestHoldingsRecords *(Utilice OAI-PMH en su lugar)*
- GetRecords
- Buscar *(Usar en cambio SRU)*
- Escanear*(Usar en cambio SRU)*
- GetAuthorityRecords
- OutputRewritablePage *(No soportado todavía)*
- OutputIntermediateFormat *(No soportado todavía)*

192.168.56.103/cgi-bin/koha/ilsdi.pl

Protocolo ILS-DI

Ejercitación

Habilitar el protocolo ILS-DI en las instancias de Koha de ejercitación

Chequear que el mismo está correctamente habilitado y verificar su securización solo a la IP del servidor donde se encuentra instalado vuFind (10.1.12.159).

Revisar la documentación en línea de vuFind en relación a la interoperabilidad con Koha.

Proponer una configuración posible para la conexión de vuFind con Koha

Determinar cuáles servicios podría llegar a implementar vuFind.

Debatir si vuFind puede o no ser una alternativa (enriquecida por el potencial de incorporación de recursos) para el OPAC de Koha.

Protocolos aceptados por KOHA

REST / OpenAPI / Swagger

¿Qué es REST?

REST, REpresentational State Transfer, es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP.

REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RPC.

REST se definió en el 2000 por Roy Fielding, coautor principal también de la especificación HTTP. Podríamos considerar REST como un framework para construir aplicaciones web respetando HTTP.

Por lo tanto REST es el tipo de arquitectura más natural y estándar para crear APIs para servicios orientados a Internet.

Fuente:
https://asiermarques.com/2013/conceptos-sobre-apis-rest/

HTTP define un conjunto de **métodos de petición** para indicar la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados *HTTP verbs*. Cada uno de ellos implementan una semántica diferente, pero algunas características similares son compartidas por un grupo de ellos: ej. un *request method* puede ser safe, idempotent, o cacheable.

GET

El método `GET` solicita una representación de un recurso específico. Las peticiones que usan el método `GET` sólo deben recuperar datos.

POST

El método `POST` se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

PUT

El modo `PUT` reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

DELETE

El método `DELETE` borra un recurso en específico.

Cuando solicitamos una URL en el explorador aplica el verbo GET

Cuando enviamos un form habitualmente sale con un verbo POST

Verbos HTTP

https://developer.mozilla.org/es/docs/Web/HTTP/Methods

JSON (JavaScript Object Notation). ¿Qué es? ¿Para qué sirve?

La notación de objetos de JavaScript (JSON) es un **formato de texto** para serialización de datos estructurados bajo los principios de ser textual, mínimo, portable y un subconjunto de JavaScript

JSON puede representar **cuatro tipos primitivos** (cadenas, números, booleanos y nulo) y dos tipos estructurados (objetos y matrices).

Una cadena es una secuencia de cero o más caracteres Unicode [UNICODE].

Un objeto es una colección desordenada de cero o más pares nombre / valor, donde un nombre es una cadena y un valor es una cadena, número, booleano, nulo, objeto o matriz.

Una matriz es una secuencia ordenada de cero o más valores.

Los términos "objeto" y "matriz" provienen de las convenciones de JavaScript.

Comparando XML & JSON

```xml
<menu>
    <header>Adobe SVG Viewer</header>
    <item action="Open" id="Open">Open</item>
    <item action="OpenNew" id="OpenNew">Open New</item>
    <separator/>
    <item action="ZoomIn" id="ZoomIn">Zoom In</item>
    <item action="ZoomOut" id="ZoomOut">Zoom Out</item>
    <item action="OriginalView" id="OriginalView">Original View</item>
    <separator/>
    <item action="Quality" id="Quality">Quality</item>
    <item action="Pause" id="Pause">Pause</item>
    <item action="Mute" id="Mute">Mute</item>
    <separator/>
    <item action="Find" id="Find">Find...</item>
    <item action="FindAgain" id="FindAgain">Find Again</item>
    <item action="Copy" id="Copy">Copy</item>
    <item action="CopyAgain" id="CopyAgain">Copy Again</item>
    <item action="CopySVG" id="CopySVG">Copy SVG</item>
    <item action="ViewSVG" id="ViewSVG">View SVG</item>
    <item action="ViewSource" id="ViewSource">View Source</item>
    <item action="SaveAs" id="SaveAs">Save As</item>
    <separator/>
    <item action="Help" id="Help">Help</item>
    <item action="About" id="About">About Adobe CVG Viewer...</item>
</menu>
```

```json
{"menu": {
    "header": "SVG Viewer",
    "items": [
        {"id": "Open"},
        {"id": "OpenNew", "label": "Open New"},
        null,
        {"id": "ZoomIn", "label": "Zoom In"},
        {"id": "ZoomOut", "label": "Zoom Out"},
        {"id": "OriginalView", "label": "Original View"},
        null,
        {"id": "Quality"},
        {"id": "Pause"},
        {"id": "Mute"},
        null,
        {"id": "Find", "label": "Find..."},
        {"id": "FindAgain", "label": "Find Again"},
        {"id": "Copy"},
        {"id": "CopyAgain", "label": "Copy Again"},
        {"id": "CopySVG", "label": "Copy SVG"},
        {"id": "ViewSVG", "label": "View SVG"},
        {"id": "ViewSource", "label": "View Source"},
        {"id": "SaveAs", "label": "Save As"},
        null,
        {"id": "Help"},
        {"id": "About", "label": "About Adobe CVG Viewer..."}
    ]
}}
```

Mini-Ejercicio

Expresar en términos de la definición de JSON (escalares, vectores y objetos) el ejemplo anterior.

Dado el siguiente JSON

[{"address1":null,"address2":null,"address3":null,"city":null,"country":null,"email":null,"fax":null,"geolocation":null,"ip":null,"library_id":"CEN","marc_org_code":null,"name":"Biblioteca Central","notes":null,"opac_info":null,"phone":null,"pickup_location":true,"postal_code":null,"reply_to_email":null,"return_path_email":null,"state":null,"url":null},{"address1":null,"address2":null,"address3":null,"city":null,"country":null,"email":null,"fax":null,"geolocation":null,"ip":null,"library_id":"POS","marc_org_code":"Bpg","name":"Biblioteca del Postgrado","notes":null,"opac_info":null,"phone":null,"pickup_location":true,"postal_code":null,"reply_to_email":null,"return_path_email":null,"state":null,"url":null}]

Buscar en la web alguna herramienta para facilitar su visualización y comprensión

What Is OpenAPI?

OpenAPI Specification (formerly Swagger Specification) is an API description format for REST APIs. An OpenAPI file allows you to describe your entire API, including:

- Available endpoints (`/users`) and operations on each endpoint (`GET /users`, `POST /users`)
- Operation parameters Input and output for each operation
- Authentication methods
- Contact information, license, terms of use and other information.

API specifications can be written in YAML or JSON. The format is easy to learn and readable to both humans and machines. The complete OpenAPI Specification can be found on GitHub: OpenAPI 3.0 Specification

Con los 4 slides de arriba, entonces tenemos que OpenAPI será una especificación de una API REST escrita usando JSON

Mini-Ejercicio (II)

Analizar la especificación de ejemplo para OpenAPI que está en

https://github.com/OAI/OpenAPI-Specification/blob/master/examples/v2.0/json/petstore-minimal.json

Determinar los paths, la estructura de datos que recibe el servicio, los códigos HTTP tanto que recibe y devuelve y la estructura de datos que devuelve

https://github.com/OAI/OpenAPI-Specification/blob/master/examples/v2.0/json/petstore-minimal.json

- About Koha
- Participation
- Development
- Documentation
- Koha Versions
- Koha & Git
- Koha Conferences
- Legal Matters
- Tips & Tricks

toolbox

- What links here
- Related changes
- Special pages
- Printable version
- Permanent link
- Browse properties

Endpoint	Decision status	Implementation status	Koha version
/account/lines	Voted		
/acquisitions/budgets			
/acquisitions/budgets/{budget_id}/funds			
/acquisitions/funds	Voted	Done (matts)	**19.05, 18.11.05**
/acquisitions/invoices			
/acquisitions/orders	Voted	SO (tcohen)	
/acquisitions/vendors	Voted	Done	**17.11**
/acquisitions/vendors/{vendor_id}/contacts			
/authorities			
/authorised_values		WIP	
/authorised_values_categories		WIP	
/biblios	Voted	Doesn't Apply (tcohen)	
/biblios/ids			
/biblios/{biblio_id}/holds			
/biblios/{biblio_id}/items			
/biblios/{biblio_id}/pickup_locations			
/checkouts	Voted	Done	**19.05**
/checkouts/{checkout_id}/allows_renewal	Voted	Done	**19.11 19.05.03**
/cities	Voted	Done	**18.05**
/holds	Voted	Done (tcohen)	**19.05**
/holds/{hold_id}/priority		SO (tcohen)	
/holds/{hold_id}/suspension	Voted	Done (tcohen)	**19.05**
/ill_backends	Voted	Assigned (josef.moravec)	
/ill_requests			
/items			
/items/{item_id}/pickup_locations			
/libraries	Voted	Done (tcohen)	**19.05, 18.11.04**
/patrons	Voted	Done	**18.05**
/patrons/{patron_id}/account	Voted	Done	**18.11, 18.05.03**

Endpoints actualmente implementados por la API REST de Koha

https://wiki.koha-community.org/wiki/REST_api_RFCs

Protocolos de autenticación soportados por KOHA

Koha offers several options for using external authentication services to allow users to sign in. In this blog post I will list the options and cover some of the basics of using different options.

LDAP

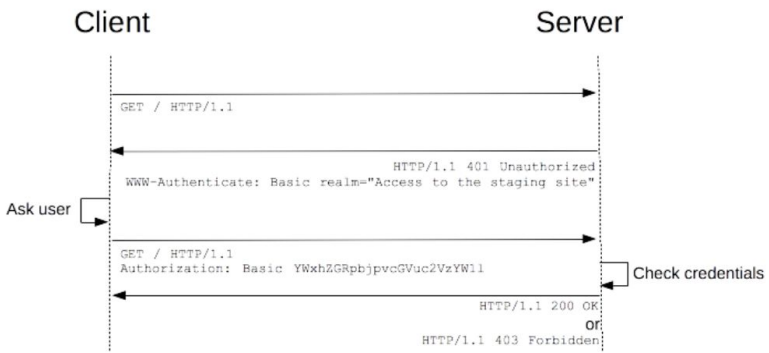
CAS

Google OAUTH

SAML

Related: Ebsco EDS/OpenAthens plugins

Para la ejercitación con la API REST de Koha vamos a usar la autenticación básica de HTTP.



Este esquema de autenticación no usa una servidor de identidad federada sino que se basa en los mismos usuarios de Koha. La combinación usuario y passwd se envía codificada. Para resultar confiable debería funcionar sobre https

Preferencias que es necesario setear en KOHA para habilitar el funcionamiento de la API REST

preferencias

▼ REST API

Preferencia	Valor
RESTBasicAuth	Habilitar ▼ Autenticación básica para el REST API.
RESTdefaultPageSize	Establezca el número predeterminado de resultados devueltos por los puntos finales de la REST API a 20 por página.
RESTOAuth2ClientCredentials	Deshabilite ▼ las credenciales de cliente OAuth2 para la API REST. Requiere tener instalado Net::OAuth2::AuthorizationServer . [EXPERIMENTAL]
RESTPublicAPI	Habilitar ▼ el /espacio de nombre público de la API.

Guarde todas las preferencias Cancelar

Con estas preferencias estaremos habilitando la API REST e indicaremos que usaremos el método de autenticación básico para consumir los servicios.

Probando la API REST

No es seguro | 192.168.0.30/api/v1/libraries

[{"address1":null,"address2":null,"address3":null,"city":null,"country":null,"email":null,"fax":null,"geolocation":null,"ip":null,"library_id":"CEN","marc_org_code":null,"name":"Central","notes":null,"opac_info":null,"phone":null,"pickup_location":true,"postal_code":null,"reply_to_email":null,"return_path_email":null,"state":null,"url":null}]

Si estamos logueados en intranet o el OPAC, los headers de autenticación se enviarán automáticamente y podremos consumir los servicios (Si usamos el mismo navegador)

No es seguro | 192.168.0.30:8080/api/v1/libraries

{"error":"Session has been expired."}

Para acceder a un listado de endpoints del API REST de Koha debemos acceder a: <OPACURL>/api/v1/.html

Las aplicaciones que diseñemos tendrán que enviar los headers de autenticación básica para poder consumir los servicios de la API. Pero cómo podemos armar un sandbox donde probemos los diferentes servicios y su impacto en Koha?

Extensión (RESTer) moz-extension://ca271eaf-1e58-42fb-b70b-e8fc6fd

Buscar

RESTer

Collection / Title

Requests

Settings

Environment

Organize

History

03:52:55
PUT http://192.168.0.237/api/v1/contrib/kitchensin...

03:52:14
GET http://192.168.0.237/api/v1/patrons

03:49:20
PUT http://192.168.56.103/api/v1/patrons/2/bother

03:49:06
GET http://192.168.56.103/api/v1/patrons/2

03:48:59
GET http://192.168.56.103/api/v1/2/patrons

Method

URL

SEND

HEADERS BODY AUTHORIZATION VARIABLES

Name

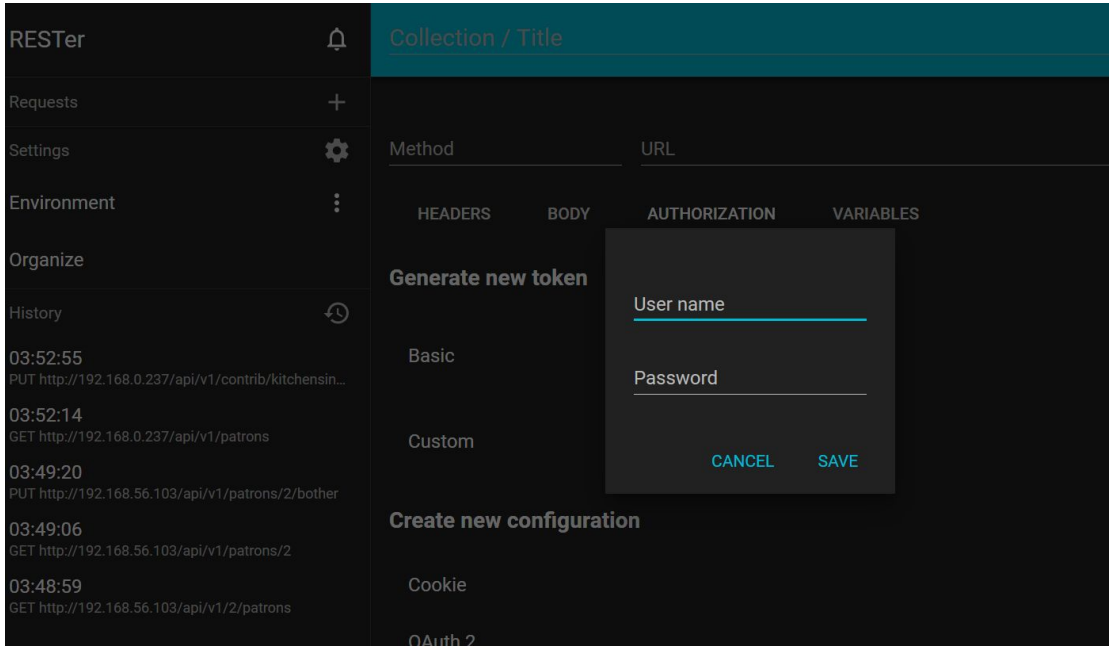
Value

RESTer: https://addons.mozilla.org/es/firefox/addon/rester/

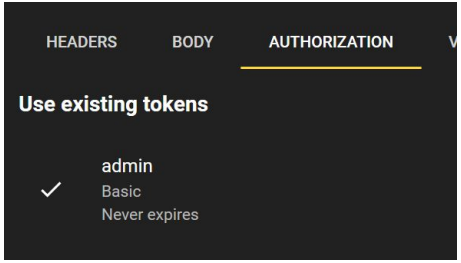
Cliente REST que nos permite consumir y debuggear un servicio

RESTer. Definiendo la autenticación básica

Con este logo abriremos la extensión de Mozilla RESTer



Crearemos un nuevo request y en la pestaña Authorization agregaremos credenciales de autenticación, para lo cual usaremos las de un usuario superlibrarian de KOHA.



RESTer. Haciendo nuestro primer request!

Method
GET

URL
http://192.168.0.30/api/v1/libraries

HEADERS BODY AUTHORIZATION VARIABLES

Use existing tokens

✓ admin
Basic
Never expires

Response 200 OK

```
Date: Sun, 29 Sep 2019 15:14:41 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Type: application/json;charset=UTF-8
Content-Length: 344
Vary: User-Agent
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```


```
1  [
2      {
3          "address1": null,
4          "address2": null,
5          "address3": null,
6          "city": null,
7          "country": null,
8          "email": null,
9          "fax": null,
10         "geolocation": null,
11         "ip": null,
12         "library_id": "CEN",
13         "marc_org_code": null,
14         "name": "Central",
15         "notes": null,
16         "opac_info": null,
17         "phone": null,
18         "pickup_location": true,
19         "postal_code": null,
```


Si bajamos desde la misma pantalla, veremos el resultado en JSON obtenido del servicio y el código HTTP devuelto, en este caso 200 (OK)

REST API

Ejercitación

- Instalar la extensión RESTer en Mozilla
- Setear la preferencia de KOHA adecuada para autenticación básica
- Definir un nuevo usuario con derechos de superlibrarian llamado API-REST
- Usando la extensión RESTer, consumir la información de las bibliotecas definidas en Koha. Obtener el formato en JSON.
- La Biblioteca Central ha definido el cardnumber como la concatenación de tipo y número de documento. Usando la extensión RESTer, obtener en JSON los datos de un usuario específico contando con su tipo y número de documento.

RESTer. Enviando datos a KOHA (Agregando datos)

RESTer

Requests

Settings

Environment

Organize

History

12:21:23
POST http://192.168.0.30/api/v1/patrons

12:20:31
POST http://192.168.0.30/api/v1/patrons

12:12:28
GET http://192.168.0.30/api/v1/libraries

03:52:55
PUT http://192.168.0.237/api/v1/contrib/kitchensin...

03:52:14
GET http://192.168.0.237/api/v1/patrons

Shortcuts - About

Collection / Title

Method
POST

URL
http://192.168.0.30/api/v1/patrons

SEND

HEADERS BODY AUTHORIZATION VARIABLES

```
1 {"cardnumber":"3738933",
2  "surname":"Giménez Grau",
3  "firstname":"Graciela",
4  "address":"Felipe González 673",
5  "city":"Lanús"
6  }
```

Response 400 Bad Request

347 B 103 ms

```
Date: Sun, 29 Sep 2019 15:23:36 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 146
Content-Type: application/json;charset=UTF-8
Vary: User-Agent
Connection: close
```

```
1  {
2      "errors": [
3          {
4              "message": "Missing property.",
5              "path": "/body/category_id"
6          },
7          {
8              "message": "Missing property.",
9              "path": "/body/library_id"
10         }
11     ],
12     "status": 400
13 }
```

Ejercicio

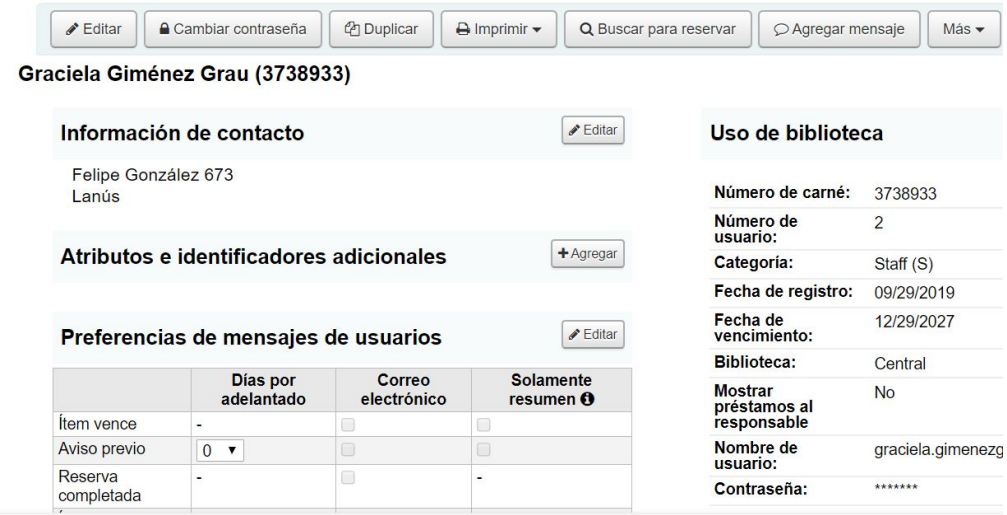
Explicar lo que ocurre aquí, el método aplicado y los datos que estarían faltando para poder completar adecuadamente el request

Y ese usuario, está efectivamente en KOHA?

Response 201 Created

```
Date: Sun, 29 Sep 2019 15:26:48 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 380
Content-Type: application/json;charset=UTF-8
Vary: User-Agent
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```


```
{
    "address": "Felipe González 673",
    "anonymized": false,
    "cardnumber": "3738933",
    "category_id": "S",
    "city": "Lanús",
    "date_enrolled": "2019-09-29T12:26:48",
    "expiry_date": "2027-12-29T12:26:48",
    "firstname": "Graciela",
    "incorrect_address": false,
    "library_id": "CEN",
    "patron_card_lost": false,
    "patron_id": 2,
    "privacy": 1,
    "restricted": false,
    "surname": "Giménez Grau",
    "userid": "graciela.gimenezgrau"
}
```

Ejercitación integrada

- Usando la extensión RESTer, crear tres nuevos usuarios de categorías estudiantes para diferentes branches. Verificar desde la interfaz de intranet de KOHA que efectivamente han sido creados y están habilitados.
- La Universidad está por implementar la app de Guaraní para los alumnos y resultaría deseable que la Biblioteca resulte integrada entre los servicios que la app presenta/dispone. Los requerimientos iniciales serían
  - Integración y notificaciones con el calendario de las devoluciones y reservas que el usuario tiene pendientes / disponibles. Funcionalidad para reservar y renovar desde la app.
  - Búsqueda en el catálogo de la biblioteca.

Ejercitación integrada

- La Facultad de Ingeniería ha diseñado un buzón automático de devoluciones, el mismo se especifica de la siguiente manera:
  - El usuario acerca al lector de código de barras integrado al buzón, el ejemplar prestado
  - Si el ejemplar corresponde a la Biblioteca de Ingeniería (solo por ahora en ella está implementado), el buzón se abre, se recibe la publicación en un compartimiento intermedio, se toma una foto digital del contenido del compartimiento y luego se almacena el ejemplar devuelto. Si no es de la Biblioteca de Ingeniería, el buzón advierte el caso y cancela la operación.
  - Si el préstamo se ha excedido por 15 días en el plazo de devolución, el buzón debe advertir al usuario que se encuentra preventivamente suspendido y que se acerque a la biblioteca en el próximo día hábil para justificar el retraso, para presentar documentación justificativa del retraso.

Ejercitación integrada

La REDIAB comienza a tratar la temática de la implementación de un protocolo de préstamo interbibliotecario.

Como mejor objetivo, el mismo debería permitir a los KOHAs conectados a la red realizar operaciones de circulación B2B de publicaciones. Para expresarlo en términos prácticos, cualquier biblioteca integrante de la red, puede realizar reservas o ser objeto de préstamo de una publicación, de acuerdo a un set de reglas pre-definidas.

Investigar en la documentación de KOHA, discutir las reglas, posibilidades y posibles implementaciones consideradas actualmente en la versión de KOHA para esta funcionalidad, incluyendo las posibilidades que la API-REST contempla de modo de informar a REDIAB.

Ejercitación integrada

La Biblioteca de la Facultad de Ingeniería decide abrir sus estanterías luego de colocar etiquetas RFID ocultas en todas sus publicaciones , con la intención de implementar autopréstamo autónomo.

- El usuario, usando los nuevos servicios de la app de Guaraní puede decidir prestarse a domicilio una publicación que encuentra de su interés recorriendo las estanterías. Aplican iguales restricciones que si estuviera en mostrador.
- Al retirarse de la Biblioteca, el usuario debe pasar por una de las terminales de salida, usando la cual un lector de RFID lee las etiquetas de los ejemplares, comprueba que estén el período de un préstamo corriente y marca las etiquetas RFID como “apta para salida”.
- Al salir de la Biblioteca, una barra de seguridad detecta (si hay) RFIDs no aptos para salida y suena la alarma.
- Por seguridad, todo RFID marcado como apto de salida al reingresar a la Biblioteca, es marcado como no apto para salida, en forma automática, incluso estando en bolsos o mochilas por las antenas diseñadas a tal fin (La biblioteca posee 1 puerta de entrada y otra de salida).

Ejercitación integrada

Para los cuatro ejercicios anteriores, es posible que sea preciso aplicar más de un protocolo de interoperabilidad de Koha, combinado con la API REST. Para todos los planteos, la ejercitación consiste en investigar el protocolo óptimo, verificar el correcto funcionamiento de los mismos con los casos planteados y realizar los documentos de especificación de los mismos, en vistas a la creación / integración de las aplicaciones.

Caso de estudio. Interoperabilidad con SIU-Guaraní