



Red Mexicana
de Repositorios
Institucionales

www.remeri.org.mx

**Paquete de
trabajo 3**

Operación y Documentación
del Portal Web del Proyecto
REMEDI

Manual Técnico

Enero 2013

HOJA EN BLANCO

Paquete de trabajo 3	Manual Técnico de Operación y Documentación del Portal Web del Proyecto REMERI
Documento desarrollado por:	José Alfredo Sánchez Huitrón, Dr Antonio Felipe Razo Rodríguez, MDI, MC Yazmín Magallanes Velázquez, MC
Revisores del documento:	Rosalina Vázquez Tapia, MTE
Aprobación requerida de:	
Nombre del proyecto:	Red Mexicana de Repositorios Institucionales REMERI
Responsable(s) del proyecto:	Rosalina Vázquez Tapia, MTE Universidad Autónoma de San Luis Potosí
Objetivo del documento:	Manual técnico de instalación y operación del portal y aplicaciones de REMERI
Fecha de Elaboración Fecha de Actualización	Diciembre 10, 2012 Enero 26, 2013
Versión del Documento	0.9.5

Manual Técnico
Red Mexicana de Repositorios Institucionales REMERI

Operación y Documentación del Portal Web del Proyecto REMERI

1. Administración de archivos y gestión de servicios en el servidor

- 1.1. Presentación
- 1.2. Características del Servidor
- 1.3. Servicios Instalados
- 1.4. Gestión de Archivos
- 1.5. Gestión de la Base de Datos
- 1.6. Políticas de Administración

2. Estructura y contenido del portal

- 2.1. Presentación
- 2.2. Secciones (galería, código)
- 2.3. Formularios
- 2.4. Consultas

3. Aplicaciones de Visualización

- 3.1. Presentación
- 3.2. Aplicaciones
- 3.3. Mantenimiento y actualización de las aplicaciones

4. Gestión de servicios de metadatos

- 3.1. Presentación
- 3.2. Configuración de los servidores
- 3.3. Consultas y pruebas
- 3.4. Validación de los servidores

1. Administración de archivos y gestión de servicios en el servidor

1.1. Presentación

La Red Mexicana de Repositorios Institucionales (REMERI), tiene por objeto crear una red interconectada de repositorios digitales de instituciones de educación superior (IES) en México, con la finalidad de integrar, difundir, preservar y dar visibilidad a su producción científica, académica y documental, así como también, incorporarse a redes o directorios de repositorios internacionales para fomentar la colaboración y apoyar el acceso y la divulgación de contenidos de acceso abierto.

El servidor y servicios del proyecto se encuentran instalados en el Centro de Datos de la Universidad Autónoma de San Luis Potosí. Este reporte muestra la relación de aplicaciones y procesos necesarios para su operación, así como una guía de instalación, configuración y administración de servicios y mantenimiento de archivos y políticas de uso.

1.2. Características del Servidor

El servidor tiene las siguientes características:

- Dell PowerEdge R720xd
 - 2 procesadores E5-2650 (16 núcleos en total)
 - 64Gb de memoria RAM a 1600 MHz en 4 DIMMs
 - Seis discos duros de 600 GB y 15,000 RPM en arreglo RAID 5 a través de la controladora PERC H710
 - Fuente de alimentación redundante de 1100 W cada una en un gabinete para rack de dos unidades

- Sistema Operativo :
 - Linux
 - OpenSUSE 12.1
 - kernel 3.1.0-1.2-desktop
 - amd-64

- Java

- Vendor: Sun Microsystems Inc.
 - Version: 1.6.0_24
 - Implementation: OpenJDK 64-Bit Server VM
 - Installation: /usr/lib64/jvm/java-1.6.0-openjdk-1.6.0/jre

- Usuario
 - Cuenta: remeri_udlap
 - Ruta: /home/remeri_udlap

- IP
 - 148.224.96.22

- Detalle del Dominio
 - Nombre de Dominio: remeri.org.mx
 - Fecha de creación: 2012-11-06
 - Fecha de expiración: 2017-11-06
 - Fecha de última modificación: 2012-11-23

1.3. Servicios Instalados

- Servidor HTTP
 - Apache Tomcat/6.0.33
 - Puerto:8080
 - Servlet Specification Version : 2.5
 - JSP version : 2.1
 - Installation: /usr/share/tomcat6

- Servidor de Base de Datos
 - eXist 1.4.2.
 - Puerto: 8090
 - Installation: /home/remeri/eXist

1.4. Gestión de Archivos

El acceso al servidor por SSH2 permite la administración de los archivos por consola o por aplicación. Y el servidor web utilizado es Apache Tomcat un servidor de aplicaciones y archivos.

- Página principal
 - Dirección: <http://www.remeri.org.mx/>
 - Ruta: /usr/share/tomcat6/webapps/ROOT/
 - Archivos:
 - Index.html
 - favicon.ico
 - remeri.ico
 - remeri_title.gif
 - remeri_web_conacyt.png

- Aplicación principal del portal
 - Dirección: <http://www.remeri.org.mx/remeri>
 - Ruta: /usr/share/tomcat6/webapps/remeri
 - Contexto:
 - META-INF
 - WEB-INF
 - lib
 - activation.jar
 - commons-fileupload-1.1-dev.jar
 - commons-io-1.0.jar
 - mail.jar
 - web.xml
 - archivos.html
 - guarda_archivo.jsp
 - index.html
 - list.jsp
 - msg.jsp
 - registro.html
 - remeri.war
 - Archivos:
 - HTML
 - pcapacitacion.html
 - pformulario.html
 - pformulariocursos.html
 - pidentidad.html
 - pinformacion.html
 - pparticipantes.html
 - pproyecto.html
 - premeri.html
 - pservicios.html

- JSP
 - pinformacion.jsp
 - pregistrocursos.jsp
 - REMERII.jsp
 - REMERIII.jsp
 - REMERIV.jsp
 - REMERIV.jsp
- JavaScript
 - jquery.touchwipe.1.1.1.js
 - netcu.js
 - jquery.cycle.all.js
- CSS
 - premeri.css
 - indixe02.css
- SWF
 - aplicacion.swf
- Imágenes
 - remeri_12_10_25_01.gif
 - remeri_title.gif
 - logo_itesm.png
 - logo_uaeh.png
 - logo_uaemex.png
 - logo_uaslp.png
 - logo_udg.png
 - logo_udlap.png
 - mapa_e.png
 - rbanner02.png
 - rbanner04.png
 - rbanner05.png
 - remeri_web_conacyt.png
 - social_links.png
 - aplicacion_circle.jpg
 - fotos_remeri_01.jpg
 - fotos_remeri_02.jpg
- Aplicación VOROSOM de visualización
 - Dirección: <http://www.remeri.org.mx/vorosom>
 - Ruta: /usr/share/tomcat6/webapps/vorosom
 - Archivos
 - colorbrewer.css

- d3.v2.js
- index.html
- ReMeRi_1_1_0_0.csv
- ReMeRi_1_1_0_0_small.csv
- voronoi.css
- voronoi.js
- Aplicación SUNGROUPS de visualización
 - Dirección: <http://www.remeri.org.mx/sungroups>
 - Ruta: /usr/share/tomcat6/webapps/sungroups
 - Archivos
 - Examples
 - css
 - base.css
 - Sunburst.css
 - Sunburst
 - img
 - jslider.plastic.png
 - aplicacion.html
 - aplicacion.js
 - aplicacionM.html
 - aplicacionV.html
 - aplicacionVM.html
 - buscar_b.png
 - check.js
 - fatras.png
 - favicon.ico
 - favicon.png
 - fdelante.png
 - flechitafecha.png
 - flechitafechase.png
 - flechitas.png
 - jquery-1.8.2.js
 - jquery-ui.css
 - jquery-ui.js
 - jquery-ui2.css
 - remeri.json
 - remeriestilo.css
 - remeriM.json
 - remeriV.json
 - styler.css
 - Extras
 - excanvas.js
 - favicon.ico
 - jit.js
 - jit-yc.js

1.5. Gestión de Base de Datos

La base de datos que utiliza el proyecto REMERI es una base de datos XML denominada eXist.

La base de datos es un servicio web que se encuentra disponible en la siguiente dirección:

- <http://www.remeri.org.mx/exist/index.xml>

La base de datos ofrece dos tipos de interfaces para su administración. Una interfaz web:

- <http://www.remeri.org.mx/exist/admin/admin.xql>

Y una aplicación web disponible en:

- <http://www.remeri.org.mx/exist/webstart/exist.jnlp>

Al momento de ingresar se utiliza el usuario de administrador o el usuario *remeri*. Como cualquier administrador de base de datos se pueden registrar usuarios, realizar respaldos y restauraciones, eliminar documentos y colecciones o agregar documentos y colecciones.

La base de datos está organizada por colecciones. Las colecciones que se utilizan en el proyecto son las siguientes:

- db/remeri
 - consulta
 - consultas, búsquedas y listados del proyecto
 - autores.xq
 - busca.xql
 - busca_id.xq
 - consulta.xql
 - consultas_autores.xq
 - lista_autores.xq
 - lista_autores_min.xq
 - lista_fechas_min.xq
 - lista_fechas.xq
 - lista_instituciones.xq
 - lista_tipos_min.xq
 - cosechador

- scripts de recolección de metadatos, un script por institución
 - harvester_GetRecord_.xq
 - harvester_GetRecord_remeri.xq
 - harvester_GetRecord_uaeh.xq
 - harvester_GetRecord_udlap.xq
 - harvester_GetRecord_uaemex.xq
 - harvester_GetRecord_itesm.xq
 - harvester_GetRecord_unam.xq
 - harvester_GetRecord_ucsj.xq
 - harvester_GetRecord_uanl.xq
 - harvester_GetRecord_uv.xq
 - harvester_GetRecord_udg.xq
- driver
 - Servidor de metadatos para LA-Referencia
 - driver_server.xq
 - driver_.xml
- integrador
 - scripts de integración de metadatos a la colección
 - aa_remeri_.xq
 - aa_remeri_uaeh.xq
 - aa_remeri_udlap.xq
 - aa_remeri_uaslp.xq
 - aa_remeri_uaemex.xq
 - aa_remeri_itesm.xq
 - aa_remeri_unam.xq
 - aa_remeri_ucsj.xq
 - aa_remeri_uanl.xq
 - aa_remeri_uv.xq
 - aa_remeri_udg.xq
- oai
 - servidor de metadatos de la colección
 - oai_server.xq
- referencia
 - integración de colecciones con estándares DRIVER/LA-Referencia, un script por institución. Un registro por institución.
 - aa_driver_.xq
 - aa_driver_uaeh.xq

- aa_driver_uanl.xq
- aa_driver_uaslp.xq
- aa_driver_udlap.xq
- aa_driver_unam_iiec.xq
- aa_driver_uv.xq
- driver_uaeh.xml
- driver_uanl.xml
- driver_uaslp.xml
- driver_udlap.xml
- driver_unam_iiec.xml
- driver_uv.xml
- repositorio
 - itesm.xml
 - uaeh.xml
 - uaemex.xml
 - uanl.xml
 - uaslp.xml
 - ucsj.xml
 - udg.xml
 - udlap.xml
 - unam-ojs.xml
 - unam_iiec.xml
- system
 - índice de la colección para la recuperación de información
 - config
 - db
 - remeri
 - remeri_indice.xconf

La ejecución de tareas puede realizarse de cualquiera de las siguientes formas:

- En la consola de la aplicación web
- Ejecutando el script correspondiente desde el administrador
- A través de una solicitud web y con parámetros eg:
 - <http://www.remeri.org.mx/exist/rest//db/remeri/consulta/busca.xq?search=UDLAP>

1.6. Políticas de Administración

La administración de los servicios, archivos y bases de datos están restringidas a aquellas personas con usuario y contraseña de administrador. Para el acceso al servidor fuera de la Universidad Autónoma de San Luis Potosí se requiere registrar la dirección IP (fija) del equipo y acceder a través de aplicaciones SSH2 (*putty, filezilla*). Se requieren conocimientos básicos de administración para levantar o reiniciar los servicios y de HTML, CSS, JavaScript y jQuery para actualizar los archivos del portal.

La administración de la base de datos está restringida a aquellas personas con acceso al servicio de eXist y a contar con el usuario y contraseña de administrador de la base de datos (*admin o remeri*). Para las tareas de administración de la base de datos el acceso está disponible desde cualquier navegador en las interfaces mencionadas. Se requieren conocimientos básicos de XML, xQuery y HTML para configurar los scripts.

2. Estructura y contenido del portal

2.1. Presentación

Estructura de archivos, menus, enlaces, secciones, aplicaciones. Formularios y librería se Java JSP, parámetros no código.

2.1. Secciones

La páginas del portal tienen las siguientes secciones:

Encabezado (título, hojas de estilo y fonts, scripts de la página y enlace a google analytics)

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Portal REMERI v.13.01.15.01 - SLP</title>
<link href='http://fonts.googleapis.com/css?family=PT+Sans:400,700' rel='stylesheet'
type='text/css'>
<link href='http://fonts.googleapis.com/css?family=Mako' rel='stylesheet'
type='text/css'>
<link rel="shortcut icon" href="remeri.ico">
<link rel="stylesheet" type="text/css" href="premeri.css">

<script type="text/javascript">

var _gaq = _gaq || [];
_gaq.push(['_setAccount', 'UA-36563906-1']);
_gaq.push(['_trackPageview']);

(function() {
var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async =
true;
ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
'.google-analytics.com/ga.js';
var s = document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(ga, s);
})();

</script>
</head>
```

Dentro del DIV "portal" está el encabezado, el menú y el contenido

```

<div id="portal">
<div id="banner">
  <div id="head">
    Portal de la Red Mexicana de Repositorios Institucionales
  </div><!--head -->
</div><!--banner -->

<div id="banner2">
  <div id="menu">
    <a href="premeri.html" style="padding-right: 15px;padding-left:15px;">
Principal</a> <a style="padding-right: 15px;padding-left:15px;"
href="pproyecto.html">Proyecto</a> <a style="padding-right: 15px;padding-
left:15px;"href="pservicios.html">Servicios</a> <a style="padding-right:
15px;padding-left:15px;"href="pcapacitacion.html">Capacitación</a> <a style="padding-right:
15px;padding-left:15px;"href="pparticipantes.html">Participantes</a> <a
style="padding-right: 15px;padding-
left:15px;"href="pinformacion.html">Información</a>
  </div><!--menu -->
</div><!--banner2 -->

<div id="banner3">
  <div id="content">
    ...
  </div>

  </div><!--content -->
</div><!--banner3 -->
...

```

Seguido del mapa de sitio y pie de página

```

...
<div id="banner4" >
  <div id="bottom" >
    <table border="0" cellpadding="2" cellspacing="1">
      <tr valign="top"><td width="25%">

        <a href="http://www.remeri.org.mx">REMERI</a> es un proyecto desarrollado por
instituciones participantes en la Comunidad de Bibliotecas Digitales de <a
href="http://www.cudi.mx/">CUDI</a> y de la Red Abierta de Bibliotecas Digitales <a
href="http://www.rabid.org.mx">RABID</a> con el financiamiento del Consejo
Nacional de Ciencia y Tecnología <a href="http://www.conacyt.gob.mx"
target="_blank">CONACYT</a><br/>
        <br/>
        <a href="http://www.itesm.mx" target="_blank">ITESM</a> - <a
href="http://www.uaeh.edu.mx" target="_blank">UAEH</a> - <a
href="http://www.uaemex.mx" target="_blank">UAEMEX</a> - <a

```

href="http://www.uaslp.mx" target="_blank">UASLP - UDG - UDLAP

```

</td>
<td width="16%" valign="top">
<ul style="padding:0px">
<li>
<div class="headline"><a href="pproyecto.html#acerca">Proyecto</a></div>
</li>
<li>
</li>
<li>
<a href="pproyecto.html#acerca"> Acerca de REMERI</a></li>
<li>
<a href="pproyecto.html#diagnostico">Diagnóstico Nacional</a>
</li>
<li>
<a href="pproyecto.html#equipo">Equipo de trabajo</a>
</li>
<li>
<a href="pproyecto.html#normativa">Normativa </a>
</li>
<li>
<a href="http://ict.udlap.mx/wikis/remeri/index.php/Main_Page"
target="_blank">Sitio de trabajo</a>
</li>
</ul>
</td>
<td width="13%">
<ul>
<li>
<div class="headline"><a href="pservicios.html">Servicios</a></div>
</li>
<li>
</li>
<li><a href="pservicios.html#busquedas">Búsquedas</a></li>
<li>
<a href="pservicios.html#consultas">Consultas</a></li>
<li><a href="pservicios.html#validaciones">Validaciones</a></li>
<li><a href="pservicios.html#estadisticas">Estadísticas</a></li>
<li></li>
<li>
<a href="pservicios.html#oai">OAI-PMH</a>
</li>
</ul>
</td>
<td width="15%">
<ul>
<li>

```



```

        <div class="headline"><a href="pcapacitacion.html">Capacitación</a></div>
    </li>
    <li>
    </li>
    <li><a href="pparticipantes.html#como">¿</a><a
href="pcapacitacion.html#dirigida">A quién va dirigida</a>?</li>
    <li>
    <a href="pcapacitacion.html#cursos">Cursos y talleres</a></li>
    <li><a href="pcapacitacion.html#agenda">Agenda de cursos</a> </li>
    <li><a href="pcapacitacion.html#bitacora">Bitácora de capacitación</a></li>
    <li><a href="pcapacitacion.html#materiales">Materiales y referencias</a></li>
</ul>
</td><td width="16%">
<ul>
    <li>
    <div class="headline"><a href="pparticipantes.html">Participantes</a></div>
    </li>
    <li>
    </li>
    <li></li>
    <li><a href="pparticipantes.html#como">¿Cómo participar?</a></li>
    <li><a href="pparticipantes.html#directorio">Directorio</a></li>
</ul>
</td><td width="15%">
<ul>
    <li>
    <div class="headline"><a href="pinformacion.html">Información</a></div>
    </li>
    <li>
    </li>
    <li><a href="pinformacion.html#sitios">Sitios y proyectos
relacionados</a></li>
    <li></li>
    <li><a href="pinformacion.html#herramientas">Herramientas</a></li>
    <li><a href="pinformacion.html#contacto">Contáctanos</a></li>
    <li><br/></li>
</ul>
</td>
</table>
</div><!--bottom-->
</div><!--banner4 -->

<div id="banner5">
    <div id="footer">
    2012-2013 REMERI- Todos los derechos reservados | Enero 15, 2013 |
Responsable del sitio: Antonio Razo | Contacto antonio.razo [at] udlap.mx
    </div><!--footer -->
</div><!--banner5 -->

</div><!--portal -->

```

Tanto en el título de la página como en el pié de página se actualiza la fecha y el autor de los cambios cada vez que se actualiza la página.

El banner de imágenes de la página principal tiene el siguiente código:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js"></script>
<script type="text/javascript" src="netcu.js"></script>
<script type="text/javascript" src="jquery.cycle.all.js"></script>
<script type="text/javascript" src="jquery.touchwipe.1.1.1.js"></script>
<script type="text/javascript">
<!--
$(document).ready(function() {
    $('#imagegallery').cycle({
        timeout: 8000,
        pause: 2,
        random: 1,
        fx: 'fade',
        next: '#next',
        prev: '#prev'
    });

    $("#imagegallery").touchwipe({
        wipeLeft: function() {
            $("#imagegallery").cycle("next");
        },
        wipeRight: function() {
            $("#imagegallery").cycle("prev");
        }
    });

    jQuery(document).keydown(function(e)
    {
        // Left arrow
        if (e.keyCode === 37) {
            $("#imagegallery").cycle("prev");
        }

        // Right arrow
        if (e.keyCode === 39) {
            $("#imagegallery").cycle("next");
        }
    })

});
// --></script>
<style media="all" type="text/css">

img.photo{
```

```
z-index: -1;
position: relative;
}
</style>
```

Dentro del DIV “contenido” se encuentra el div contenedor de la galería, se pueden agregar tantas fotos como se requieran:

```
<div id="inside">
  <div id="imagegallery">
    
    
  </div>
</div>
```

2.1. Formularios

Para procesar formularios se utiliza JSP y la librería de Java Mail API. Se envía un correo electrónico y adicionalmente se guarda un archivo de registro en el servidor.

Código JavaScript para validar formulario web :

```
<script type="text/JavaScript">
<!--
function MM_findObj(n, d) //v4.01
  var p,i,x; if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_validateForm() //v4.0
  var i,p,q,nm,test,num,min,max,errors="",args=MM_validateForm.arguments;
  for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=MM_findObj(args[i]);
    if (val) { nm=val.name; if ((val=val.value)!="") {
      if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
        if (p<1 || p==(val.length-1)) errors+='- '+nm+' debe ser una direccion de correo
valida.\n';
      } else if (test!='R') { num = parseFloat(val);
        if (isNaN(val)) errors+='- '+nm+' must contain a number.\n';
        if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
          min=test.substring(8,p); max=test.substring(p+1);
          if (num<min || max<num) errors+='- '+nm+' must contain a number between
'+min+' and '+max+'.\n';
        }
      }
    }
  }
}
```

```

    } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' es requisito.\n'; }
  } if (errors) alert('Los siguientes campos son obligatorios:\n'+errors);
  document.MM_returnValue = (errors == "");
}
//-->
</script>

```

Formulario de comunicación dentro del DIV “contenido” :

```

<div id="form" align="right" >
  <form accept-charset="UTF-8" method="get" id="searchform"
action="REMERIII.jsp">
  <div><input name="search" id="search" size="40" type="text"><input
value="Indixe" type="submit"></div></form>
  <i>Buscar en las colecciones</i>
  </div>
  <br/>
  <hr/>
  <p> Para información de la Red Mexicana de Repositorios Institucionales, le
pedimos llenar el siguiente formulario:
  <form accept-charset="ISO-8859-1" method="post" name="registro"
action="pinformacion.jsp">
  <table width="90%" border="0" align="center">
  <tr>
  <td width="45%"><div align="right"><span class="Estilo3">*</span>Nombre
completo:</div></td>
  <td width="55%"><div align="left">
  <input name="nombre" type="text" id="nombre" size="40" />
  </div></td>
  </tr>
  <tr>
  <td><div align="right"><span class="Estilo3">*</span>Correo electr&oacute;nico:
</div></td>
  <td><div align="left">
  <input name="correo" type="text" id="correo" size="30" />
  </div></td>
  </tr>
  <tr>
  <td><div align="right">Instituci&oacute;n: </div></td>
  <td><div align="left">
  <input name="institucion" type="text" id="institucion" size="40" />
  </div></td>
  </tr>
  <tr>
  <td></td>
  <td></td>
  </tr>
  <tr>
  <td></td>
  <td></td>
  </tr>
  </table>

```

```

</tr>
<tr>
<td><div align="right">Motivo:</div></td>
<td><p align="left">
<select name="motivo" id="motivo">
<option value="Capacitación" >Capacitación</option>
<option value="Participación como miembro" >Participación como
miembro</option>
<option value="Información del proyecto" selected="selected">Información del
proyecto</option>
<option value="Información de los servicios" >Información de los
servicios</option>
</select>
</td>
</tr>
<tr>
<td><div align="right">Explique su interés<br/> en la red</div></td>
<td>
<textarea name="interes" id="interes" cols="60" rows="5"></textarea></td>
</tr>
<tr>
<td colspan="2"><div align="center">
<input name="Send" type="submit" id="Send"
onclick="MM_validateForm('nombre','R','correo','RisEmail');return
document.MM_returnValue" value="Enviar Solicitud" />
</div></td>
</tr>
<tr>
<td colspan="2">&nbsp;</td>
</tr>
<tr>
<td colspan="2"><div align="right"><span class="Estilo3">*</span><em>indica
datos obligatorios </em></div></td>
</tr>
</table>
</form>

```

Formulario y registro de información

```

<%@ page import="java.io.*,java.util.*,javax.mail.*"%>
<%@ page import="javax.mail.internet.*,javax.activation.*"%>
<%@ page import="javax.servlet.http.*,javax.servlet.*" %>
<%@ page import="java.io.*" %>
<%@ page import="java.text.DateFormat" %>
<%@ page import="java.text.ParseException" %>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ page import="java.util.Calendar" %>
<%@ page import="java.util.Date" %>
<%@ page import="java.util.GregorianCalendar" %>

```

```

<%@ page contentType="text/html; charset=UTF-8" %>

<%
String result;

String from = "yyy@udlap.mx";
String host = "localhost";
Properties properties = System.getProperties();
properties.setProperty("mail.smtp.host", host);
Session mailSession = Session.getDefaultInstance(properties);

String[] mailAddressTo=
{"aaa@gmail.com", "bbb@gmail.com", "ccc@gmail.com", "ddd@uaslp.mx"};

InternetAddress[] mailAddress_BBC = new InternetAddress [mailAddressTo.length] ;
for(int i=0;i<mailAddressTo.length;i++){
mailAddress_BBC[i] = new InternetAddress(mailAddressTo[i]);
}

try{

String nombre=request.getParameter("nombre");
String correo=request.getParameter("correo");
String institucion=request.getParameter("institucion");
String motivo=request.getParameter("motivo");
String interes=request.getParameter("interes");
String nl="\n";
String co=" ";
String br="<br/>";

MimeMessage message = new MimeMessage(mailSession);
message.setHeader("Content-Type", "text/plain; charset=UTF-8");

message.setFrom(new InternetAddress(from));
message.addRecipient(Message.RecipientType.TO,
new InternetAddress(correo));

message.setRecipients(Message.RecipientType.BCC, mailAddress_BBC);

Date ahora = new Date();
SimpleDateFormat formato_fecha = new SimpleDateFormat("dd-MM-yyyy");
SimpleDateFormat formato_hora = new SimpleDateFormat("hh:mm:ss");
String fecha=formato_fecha.format(ahora);
String hora=formato_hora.format(ahora);

String mensaje=
fecha+nl+hora+nl+nombre+nl+correo+nl+institucion+nl+motivo+nl+interes;
String mensaje2=
fecha+co+hora+co+nombre+co+correo+co+institucion+co+motivo+co+interes;

```

```

String mensaje3=mensaje+nl+nl+mensaje2;
String mensaje4=
fecha+br+hora+br+nombre+br+correo+br+institucion+br+motivo+br+interes;

String separator=System.getProperty("file.separator");
String cPath=application.getRealPath(request.getServletPath());
int buscad=cPath.lastIndexOf(separator)+1;
String filePath=cPath.substring(0,buscad)+separator;
String str = mensaje2+nl;
String nameOfTextFile = "informacion.txt";

FileWriter fstream = new FileWriter(filePath+nameOfTextFile,true);
BufferedWriter outS = new BufferedWriter(fstream);
outS.write(str);
outS.close();
message.setSubject("REMERI - Solicitud de información \""+motivo+"\"", "UTF-8");

String saludo="Solicitud de información (SLP)" +nl;
String despedida=nl+nl+"Proyecto REMERI";

message.setText(saludo+mensaje+despedida,"UTF-8");

Transport.send(message);
result = "SLP - Envío de solicitud exitoso."+br+mensaje4;

}catch (MessagingException mex) {
mex.printStackTrace();
result = "SLP - Error: no fue posible envío de solicitud.";
}
catch(IOException e) {
out.println(e.getMessage());
result = "SLP - Error: no fue posible escribir el registro.";
}
}
%>

```

El envío de correo se utiliza en los siguientes archivos :

- JSP
 - pinformacion.jsp
 - pregistrocursos.jsp

2.4. Consultas web

- Las consultas web se realizan utilizando los siguientes archivos JSP que encapsulan el diseño web e incorporan llamados a las respectivas consultas :

- REMERII.jsp
 - Interfaz de
 - busca_id.xq
 - parámetro
 - id
 - resultado metadatos del objeto

- REMERIII.jsp
 - Interfaz de
 - busca.xql
 - parámetros
 - search
 - el término a buscar
 - ind
 - posición de resultados
 - step
 - número de resultados
 - order
 - tipo de ordenamiento
 - 1
 - Relevancia
 - 2
 - Fecha
 - 3
 - Autor
 - 4
 - Siglas (institución)
 - resultado ordenado por relevancia

- REMERIIIV.jsp
 - Interfaz de
 - consulta.xql
 - parámetros
 - search
 - el término a buscar
 - ind
 - posición de resultados
 - step
 - número de resultados
 - order
 - tipo de ordenamiento
 - 1
 - Relevancia
 - 2
 - Fecha

- 3
 - Autor
 - 4
 - Siglas (institución)
 - type
 - tipo de consulta
 - 1
 - Por fecha eg “1999” para la década se utiliza “199”
 - 2
 - Por autor con nombre completo
 - 3
 - Por tipo de documento eg. “Tesis”
 - 4
 - Por institución

Todos estos archivos, procesan los parámetros (los transforman en caso de ser necesario) y los anexan a un *iframe* que llamara a su correspondiente consulta de la base de datos. Ejemplo de REMERII.jsp

...

```
<%
String ids="";

try{
ids= request.getParameter("id");

}catch(Exception e){
//out.println(e);
}
ids=ids.replaceAll(".txt", "");
ids=ids.replaceAll("_", "/");
ids=ids.replaceAll("_", ":");

%>
---
<iframe frameborder="0" width="670" height="400"
src=" ../exist/rest/db/remeri/consulta/busca_id.xq?id=<%=ids%>"></iframe>
----
```

3. Aplicaciones de Visualización

3.1. Presentación

Buscar y explorar en grandes colecciones de información se ha convertido en un área de investigación muy interesante. Esto se debe a que se necesita mostrar al usuario la información que desea, pero sin saturarlo de elementos, es decir, exponerle lo esencial aunque siempre dándole espacio para explorar dentro de los datos que se le otorgan.

Uno de los esquemas de visualización que retoma el proyecto de REMERI, es el conocido como “Sunburst” basado en “TreeMap” pero que a diferencia de éste, se utilizan sectores de un plano circular.

Por otro lado, también se optó por el uso de mapas auto-organizados, los cuales proveen un panorama general de la colección y clasifican los documentos en categorías con contenidos similares. Tal organización puede ayudar a los usuarios en sus tareas de exploración y descubrimiento de conocimiento. Para ello, se diseñó una interfaz que implementa la visualización de distribuciones de clases en mapas auto-organizados y al mismo tiempo se hace uso de la visualización metáfora de cielo. Ambas visualizaciones permiten crear sobre ellas un diagrama de Voronoi. Esta interfaz permite la navegación a través de cada uno de los mapas generados en el paso anterior.

3.2. Aplicaciones de Visualización

Vorosom

Se propone un método de clasificación semi-automatizada para generar mapas auto-organizados. Esto es posible debido a que se cuenta con títulos y descripciones de los recursos. De esta clasificación se propone un modelo que muestra las colecciones en regiones de voronoi.

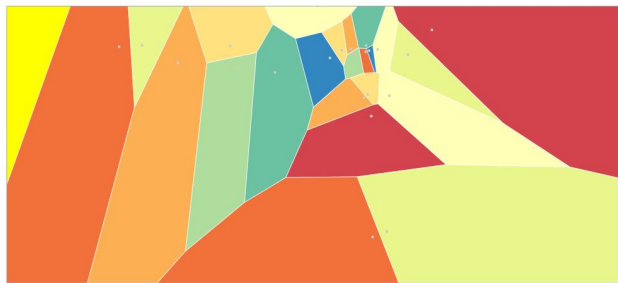


Figura 2. Interfaz de Vorosom.

A continuación se describe el listado de archivos necesarios para su ejecución:

- colorbrewer.css
- d3.v2.js
- index.html
- ReMeRi_1_1_0_0.csv
- ReMeRi_1_1_0_0_small.csv
- voronoi.css
- voronoi.js

Sungroups

Otra de las propuestas está basada en la representación "Sunburst" de John Stasko, en la cual se presenta un círculo con una serie de sectores que agrupan por palabras clave a las diferentes publicaciones existentes dentro de REMERI.

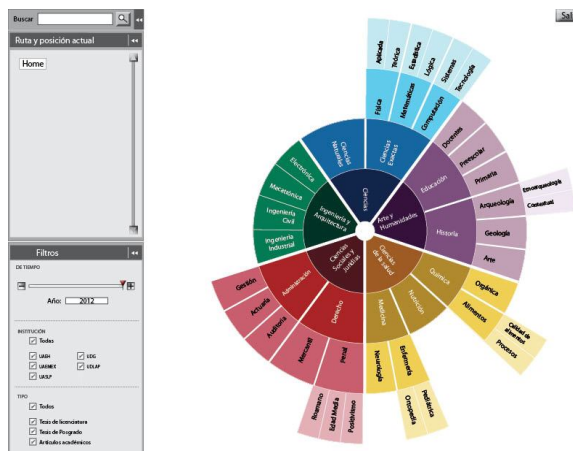


Figura 2. Interfaz de Sungroups.

A continuación se describe el listado de archivos necesarios para su ejecución:

- Examples
 - css
 - base.css
 - Sunburst.css
 - Sunburst
 - img
 - jsliider.plastic.png
 - aplicacion.html
 - aplicacion.js
 - aplicacionM.html
 - aplicacionV.html
 - aplicacionVM.html
 - buscar_b.png

- check.js
- fatras.png
- favicon.ico
- favicon.png
- fdelante.png
- flechitafecha.png
- flechitafechase.png
- flechitas.png
- jquery-1.8.2.js
- jquery-ui.css
- jquery-ui.js
- jquery-ui2.css
- remeri.json
- remeriestilo.css
- remeriM.json
- remeriV.json
- styler.css
- Extras
 - excanvas.js
 - favicon.ico
- jit.js
- jit-yc.js

3.4. Mantenimiento y actualización de las aplicaciones

Los datos son recuperados de la base de datos eXist y se convierten en formato de texto plano para su posterior procesamiento.

Se hace uso de tres atributos de Dublin Core para la generación de los archivos .txt: Título (Title), Descripción (Description) y Clasificación (Subject). Dichos atributos se escogieron con base en las necesidades de clasificación automática de la visualización.

Es posible implementar consultas a la base de datos para recuperar colecciones específicas y visualizar su contenido únicamente. Esto es de importancia ya que a partir de consultas específicas se crean los filtros de las visualizaciones.

Todo el código para el procesamiento de la información se encuentra en una carpeta llamada “vorosom”. Esta carpeta contiene varios paquetes, en donde cada paquete define tres pasos que se muestran en la Figura 1. Cada paso define una etapa: 1) pre-procesamiento, 2) construcción de los mapas auto-organizados y 3) generación de la visualización a partir de los mapas auto-organizados

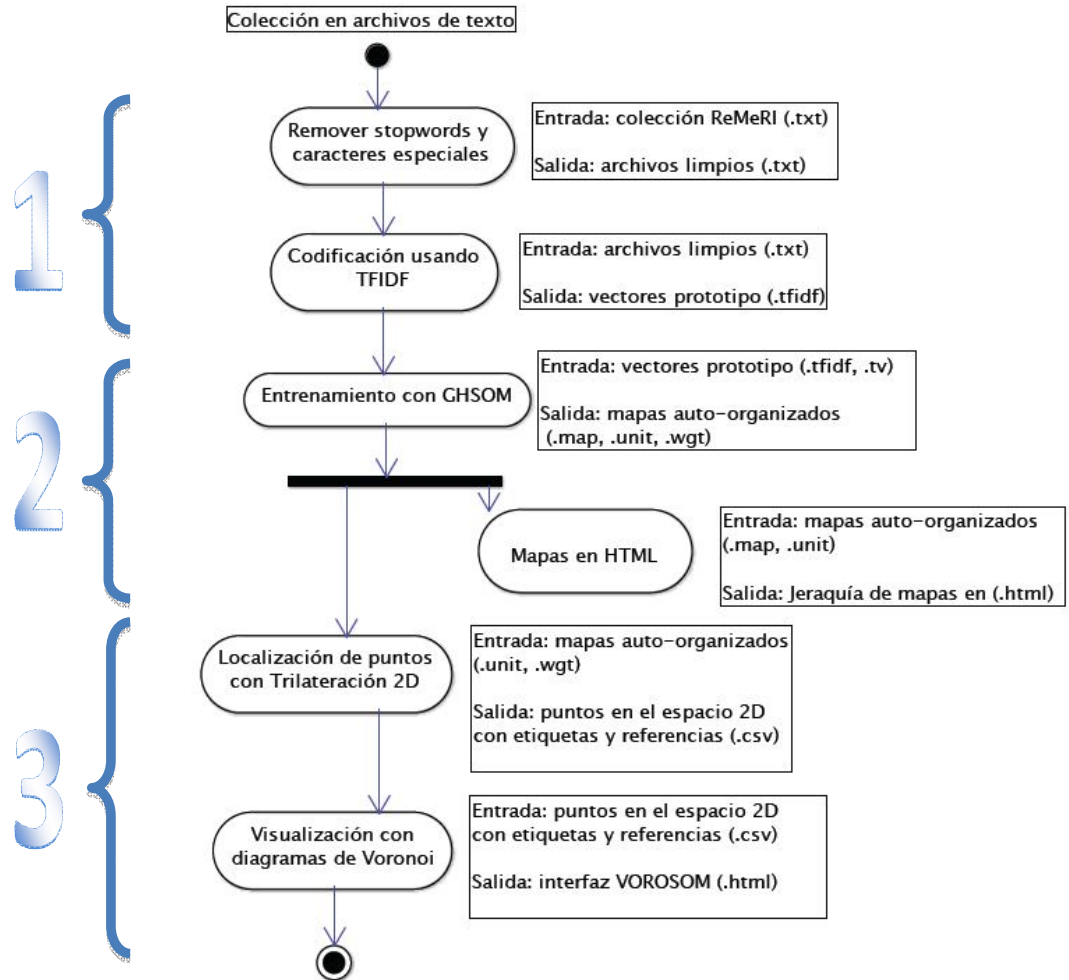


Figura 1. Pasos para la construcción de la visualización VOROSOM

Pre-procesamiento y codificación de los documentos

Para el procesamiento y codificación de documentos se utilizó el lenguaje de programación Java. Se utilizaron dos paquetes:

El primero para la eliminación de palabras *stopword* (paquete stopwords). El resultado de la ejecución de este código es la creación de un directorio de archivos .txt que serán usados para la codificación sin stopwords y sin caracteres especiales.

El segundo para la generación de los vectores prototipo usando TFIDF (paquete somlib). El resultado de esta etapa fue la creación de un par de archivos (.tfxidf y .tv) con vectores prototipo por cada uno de los documentos de la colección. Este archivo describe los términos más importantes de cada documento en formato numérico.

Construcción de los mapas auto-organizados

Para la implementación del algoritmo de redes neuronales, se hizo uso de un algoritmo ya implementado en lenguaje C por la Universidad Tecnológica de Viena (TU Wien). Este algoritmo es en específico el GHSOM que permite crear mapas auto-organizados de manera dinámica (en tamaño y número) creando una jerarquía de mapas. La entrada al algoritmo es cada uno de los archivos generados por el código que implementa TFIDF generado en la etapa de pre-procesamiento y codificación de documentos. El resultado del algoritmo GHSOM es una jerarquía de rejillas de neuronas, en donde cada neurona cuenta con su vector prototipo.

Generación de la visualización a partir de los mapas auto-organizados

Posterior a la generación de las neuronas es preciso determinar la ubicación espacial para su uso en la visualización. A partir de los vectores prototipo de las neuronas resultado de GHSOM, se desarrolló un código en Java con un algoritmo de trilateración espacial en 2-D. El resultado del algoritmo es la ubicación espacial de las neuronas con base en la cercanía de las características entre ellas (determinadas por los vectores prototipo).

Para la ubicación en el espacio 2D de los puntos se creó el paquete puntos_voronoi que incluye una serie de clases que mapean los valores de los archivos .wgt al espacio 2D y posteriormente crean un archivo .csv que incluye estos puntos relacionados con sus etiquetas y mapas respectivos usando como base los archivos .unit creados en la etapa anterior.

El resultado de esta etapa es un conjunto de puntos en el espacio 2-D que son usados para la generación los elementos gráficos de la visualización, en donde cada uno de los puntos está relacionado con sus etiquetas y con mapas subsecuentes.

Ejecución del código (paso a paso)

1. Copiar el resultado de la cosecha (archivos txt) a una carpeta llamada "files" (si no existe crearla).

```
mkdir /home/user_name/vorosom/files
```

2. Abrir una terminal y cambiarse al directorio del paquete stopwords:

```
cd /home/user_name/vorosom/stopwords
```

3. Ejecutar la clase `DirListing`, que tiene dos argumentos de entrada, con el siguiente comando:

```
java DirListing /home/user_name/vorosom/files  
/home/user_name/vorosom/tfidf/files_cleaned
```

4. Cambiarse al directorio `tfidf`

```
cd /home/user_name/vorosom/tfidf
```

5. Ejecutar el script **vector_creation_script** con los parámetros adecuados para la colección. El ejemplo que estamos usando, es de la colección ReMeRI con 24,793 documentos.

```
./vector_creation_script vectores ./files_cleaned
```

6. El resultado es la creación de dos archivos:

```
vectores.tfidf y vectores.tv
```

7. Cambiarse al directorio en donde esta `ghsom`

```
cd /home/user_name/vorosom/ghsom
```

8. Ejecutar el comando `ghsom` dando como entrada el archivo `.prop` en donde se detallan las propiedades del `ghsom`.

```
./ghsom ./properties/som_ghsom1.prop
```

9. El resultado es la creación de un directorio `output` con los archivos `.html`, `.wgt` y `.unit`

10. Cambiarse al directorio `vertices_voronoi`

```
cd /home/user_name/vorosom/vértices_voronoi
```

11. Ejecutar la clase `DirListingFilter` que toma como parámetro de entrada el directorio `output` de la carpeta `ghsom`

```
java DirListingFilter /home/user_name/vorosom/ghsom/output
```

12. Ejecutar la clase `ReMeRicsv` que toma como dos parámetros de entrada: el primero la carpeta `puntos` generada en el paso anterior y el segundo el directorio `output` de la carpeta `ghsom`

```
java ReMeRicsv puntos /home/user_name/vorosom/ghsom/output
```

13. El resultado es la creación de una carpeta llamada `ReMeRi` con los `puntos`, `etiquetas` y `mapas` asociados en un archivo `ReMeRi.csv`

14. Copiar el archivo resultante del paso anterior (`ReMeRi_1_1_0_0.csv`) a la carpeta `web`.

15. Copiar toda la carpeta `web` al servidor

Mayor información ver anexos: "Vorosom_paso_a_paso".

4. Administración y gestión de servicios de metadatos

4.1. Presentación

El protocolo OAI-PMH permite mostrar la información de una colección de manera estructurada usando el estándar Dublin Core (DC). REMERI tiene implementados dos servidores de metadatos, el primero muestra la integración de varios repositorios institucionales y temáticos. El segundo es la integración de repositorios institucionales que cumplen con los lineamientos de LA-Referencia y DRIVER 2.0.

Metadatos (OAI_DC Type)	Metadatos REQUISITO para cumplir con REMERI	Metadatos REQUISITO para cumplir con directrices de DRIVER 2.0 (2008) y con proyecto internacionales*
Title	Title	Title
Identifier	Identifier	Identifier
Description		(No obligatorio; y se recomienda en español e inglés)
Subject		
Creator	Creator	Creator
Type	Type	Type
Date	Date	Date
Rights		Rights
Publisher		
Language		Language
Format		
Coverage		
Contributor		
Source		
Relation		

4.2. Configuración de los servidores

El servidor para OAI-PMH se encuentra en la siguiente ruta:

- /db/remeri/oai

El servidor de metadatos OAI-PMH es el siguiente script:

- oai_server.xq

Se deben de configurar los siguientes parámetros para el servidor de metadatos:

```
declare variable $base-url      :=
'http://www.remeri.org.mx/exist/rest/db/remeri/oai/oai_server.xq';
declare variable $repository-name := 'Red Mexicana de Repositorios Institucionales';
declare variable $admin-email    := 'yyy@gmail.com';
declare variable $hits-per-page  := 300;
declare variable $earliest-datestamp := '1990-01-01T00:00:00Z' cast as xs:dateTime;
declare variable $db-paths      := '/remeri/repositorio';
declare variable $oai-domain    := 'remeri.org.mx';
declare variable $id-scheme     := 'oai';
```

```
...
declare function local:get-docs() {
  let $docs :=
    (:for $path in $db-paths
    :)
  let $root := collection('/remeri/repositorio)
  return
    $root//registro
...

```

Este servicio consulta los metadatos ya incorporados al repositorio.

Recolección, integración y búsquedas sobre 10 repositorios institucionales y temáticos (enero 2013)

UDLAP	4,257	Tesis de grado y posgrado
UASLP	829	Tesis de grado y posgrado
UAEH	653	Tesis de grado y posgrado
ITESM	562	Objetos de aprendizaje
UDG	116	Revista electrónica
UAEMEX	43	Repositorio temático
UNAM	14,990	Repositorio RUIIEc
UANL	2,420	Repositorio temático
UCSJ	3,001	Repositorio temático
UV:	12,103	Repositorio temático
Total:	38,970	

El servidor para DRIVER se encuentra en la siguiente ruta:

- /db/remeri/driver

El servidor de metadatos OAI-PMH es el siguiente script:

- driver_server.xq

Para integrar las colecciones para este servicio se utilizan los siguientes scripts que tienen consultas particulares para recuperar específicamente artículos, libros, tesis de maestría o de doctorado de acuerdo a los tipos de datos integrados, cada servidor es diferente por lo que hay que configurar cada script de manera particular de acuerdo a las especificaciones de LA-Referencia:

- /db/remeri/referencia
 - aa_driver_uaeh.xq
 - aa_driver_uanl.xq
 - aa_driver_uaslp.xq
 - aa_driver_udlap.xq
 - aa_driver_unam_iiiec.xq
 - aa_driver_uv.xq

Esto genera los correspondientes documentos que contienen los metadatos normalizados:

- /db/remeri/referencia
 - driver_uaeh.xml
 - driver_uanl.xml
 - driver_uaslp.xml
 - driver_udlap.xml
 - driver_unam_iiiec.xml
 - driver_uv.xml

Para validar el total de registros incorporados por institución se utiliza el siguiente script:

```
declare namespace dc="http://purl.org/dc/elements/1.1/";
declare namespace oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/";
declare namespace oai-pmh="http://www.openarchives.org/OAI/2.0/";
let $rec := collection('/db/remeri/driver')//registro[siglas="UAEH"]
return
$rec
```

Este servidor tiene las siguientes características de acuerdo a las especificaciones de LA-Referencia, además del tipo de documento:

```
...
<dc:source>instname:{$_record/institucion[1]/text()}</dc:source>
<dc:source>reponame:Red Mexicana de Repositorios Institucionales</dc:source>
<dc:rights>Acceso Abierto</dc:rights>
...
```

Se deben de configurar los siguientes parámetros para el servidor de metadatos:

```
(: configurable variables :)
declare variable $base-url      :=
http://www.remeri.org.mx/exist/rest/db/remeri/driver/driver_server.xq';
declare variable $repository-name := 'Red Mexicana de Repositorios Institucionales';
declare variable $admin-email    := 'yyy@gmail.com';
declare variable $hits-per-page  := 100;
declare variable $earliest-datestamp := '1990-01-01T00:00:00Z' cast as xs:dateTime;
declare variable $db-paths       := '/remeri/referencia;
declare variable $oai-domain     := 'remeri.org.mx';
declare variable $id-scheme      := 'oai';
...
declare function local:get-docs() {
  let $docs :=
    (:for $path in $db-paths
    :)
  let $root := collection('/remeri/referencia')
  return
    $root//registro
...

```

Recolección selectiva de libros, artículos, tesis de maestría y doctorado en 6 repositorios institucionales al 13 de Enero

- **3,696 registros**
- De las siguientes fuentes
 - UASLP 493 tesis de posgrado
 - UDLAP 512 tesis de posgrado
 - UAEH 75 tesis de posgrado
 - UV 297 tesis de posgrado
 - UANL 2,123 tesis de posgrado y artículos
 - UNAM 156 libros

3.3. Consultas y pruebas

Cada servidor está configurado para contestar a cada uno de los verbos de OAI-PMH de su correspondiente colección

Dirección de servidor OAI-PMH:

- http://www.remeri.org.mx/exist/rest/db/remeri/oai/oai_server.xq

Identify

- http://www.remeri.org.mx/exist/rest/db/remeri/oai/oai_server.xq?verb=Identify

ListIdentifiers

- http://www.remeri.org.mx/exist/rest/db/remeri/oai/oai_server.xq?verb=ListIdentifiers&metadataPrefix=oai_dc

ListRecords

- http://www.remeri.org.mx/exist/rest/db/remeri/oai/oai_server.xq?verb=ListRecords&metadataPrefix=oai_dc

Servidor DRIVER-LA Referencia:

Dirección de servidor OAI-PMH:

- http://www.remeri.org.mx/exist/rest/db/remeri/driver/driver_server.xq

Identify

- http://www.remeri.org.mx/exist/rest/db/remeri/driver/driver_server.xq?verb=Identify

ListIdentifiers

- http://www.remeri.org.mx/exist/rest/db/remeri/driver/driver_server.xq?verb=ListIdentifiers&metadataPrefix=oai_dc

ListRecords

- http://www.remeri.org.mx/exist/rest/db/remeri/driver/driver_server.xq?verb=ListRecords&metadataPrefix=oai_dc

3.4. Validación de los servidores

La validación sintáctica se realiza con la validación por Schema de la respuesta del servidor OAI-PMH. Esta validación se asegura que el servidor responda de manera correcta utilizando la sintaxis y el vocabulario definido por el Schema, además de documentos bien formados. Cada vez que se incorpore un repositorio a la colección se debe de realizar la validación del servidor para verificar posibles inconsistencias.

Link para validar sintaxis de servicios Valida OAI

- <http://www.remeri.org.mx/validaOAI/>

Existen otros servicios de validación externos

- OAI-PMH Data provider register
<http://www.openarchives.org/data/registerasprovider.html>
- Driver Repository Validator and Registration Service
<http://validator.driver.research-infrastructures.eu>

4.5. Políticas de Administración

Los servidores de metadatos permiten la interoperabilidad con otros proyectos. El principal parámetro a considerar al momento de configurar los servidores es el tamaño del *token* o número de resultados por consulta. DRIVER establece entre 100-500 registros por consulta, este es un parámetro que se puede configurar en el script correspondiente.