

Generando datos XML a partir de bases de datos relacionales

Lic. Silvina Migani, Lic. Cristina Vera, Ing. Carlos Correa, Lic. Liliana Romera

Departamento de Informática / Facultad de Ciencias Exactas, Físicas y Naturales / UNSJ

Av. Ignacio de la Roza 590 (O)

Teléfonos: 4260353 - 4260355

silvina.migani@gmail.com; civera2@yahoo.com.ar; carcorrea@hotmail.com.ar; liliromera@yahoo.com.ar

Resumen

El lenguaje XML, los lenguajes que permiten manipular datos XML, y su impacto en el mundo de las bases de datos, es el área donde este proyecto se desarrolla. Surge como una iniciativa de docentes del área bases de datos, con la finalidad de profundizar en el estudio de XML y experimentar motores de bases de datos que lo soportan.

Palabras clave: Bases de Datos – XML

Contexto

“XML: TÉCNICAS DE GESTIÓN E INTERCAMBIO DE DATOS” - 21/ E915, es el proyecto marco de este trabajo. Fue Aprobado por el CICITCA en Octubre de 2011.

Introducción

La aparición de Internet y su creciente utilización provocó muchos cambios en los sistemas informáticos, uno de ellos relacionados a la necesidad de intercambiar información. El lenguaje XML (Extended Markup Language) es el formato ideal para esos flujos de información. Cuando XML se usa en estos escenarios, los motores de bases de datos se encuentran con la necesidad de organizar, manipular y consultar este nuevo tipo de datos. Consecuentemente, surgen los Lenguajes XPath (XML Path Language), XSLT (eXtensible Stylesheet Language Transformations), XQuery (XML Query Language) y la extensión de SQL, SQL/XML.

1. Datos y Documentos XML

XML, al igual que el lenguaje de marcas de hipertexto HTML (Hyper Text Markup Language), sobre el que

está basado la World Wide Web; tiene sus raíces en la gestión de documentos y está derivado de un lenguaje para estructurar grandes documentos, conocido como lenguaje estándar generalizado de marcas SGML (Standard Generalized Markup Language) [3]. Sin embargo, la diferencia principal entre XML y HTML es que el primero fue diseñado para transportar y almacenar datos, mientras que el último para mostrar datos.

Un archivo escrito en XML, además de proporcionar información, se describe asimismo; permitiendo al receptor entender la información recibida y, por lo tanto, procesarla. Tiene el formato de un archivo de texto plano, lo que facilita enormemente la transferencia de información, logrando independencia con respecto a diferentes plataformas.

El constructor principal de un documento XML es el elemento. Un elemento es simplemente un par de etiquetas de inicio y cierre y todo el texto que aparece entre ellas [3].

a) Documento XML bien formado

El requerimiento para que un documento XML sea bien formado es que comience con una declaración que indique que es XML, y que posea un sólo elemento raíz que contenga el cuerpo completo del mismo [2]. La estructura interna dentro de ese elemento raíz es totalmente libre, simplemente se debe respetar el correcto anidamiento de las etiquetas de inicio y cierre.

b) Documento XML válido

Aunque la estructura de un documento XML en principio es flexible, siempre y cuando se ajuste a las reglas mencionadas en el punto anterior, existen situaciones donde es conveniente que se adecúe a una estructura determinada. Generalmente esto sucede cuando los documentos deben ser procesados automáticamente [2].

son las DTDs (Document Type Definition) y los Esquemas XML. Un documento XML es válido cuando, además de estar bien formado, se ajusta a la estructura y restricciones definidas en su correspondiente DTD o Esquema XML. Si bien ambas alternativas de validación son permitidas, la segunda es más apropiada por varias razones. Por un lado soporta espacios de nombres y, por otro, permite especificar más ajustadamente la estructura, en cuanto a tipos de datos, cardinalidad, etc. Además, un Esquema XML es un documento XML propiamente dicho.

Schema XML 1.0 es recomendación del W3C desde mayo del 2001 [4]. En ella se define el lenguaje utilizado para la definición de los Esquemas XML.

2. Manipulación de datos XML

Usar datos XML requiere no sólo la capacidad de estructurarlos y almacenarlos, sino también manipularlos. Con esa finalidad, surgieron los lenguajes XPath, XQuery, XSLT y SQL/XML¹.

a) XPath

Es recomendación del W3C desde Noviembre del año 1999. Fue diseñado para ser usado por XSLT y XPointer. Su propósito es direccionar partes de un documento XML [5]. Para ello, usa expresiones de camino que permiten navegar dentro de los documentos XML, seleccionando nodos o conjuntos de nodos.

Cada expresión comprende varios pasos, los cuales pueden separarse mediante "/" o "//". Cada paso puede tener un eje, un filtro y cero o más predicados. Además, dentro del lenguaje se incluyen funciones para datos string, numéricos, fecha, booleanos, entre otros.

b) XQuery

El W3C desarrolló XQuery como el lenguaje de consulta estándar para XML. Tiene como finalidad encontrar y extraer elementos y/o atributos de documentos XML[6].

XQuery permite resolver consultas más complejas y de mayor potencia semántica que XPath. Comparte con XPath, el mismo modelo de datos y las mismas funciones y operadores. Una consulta en XQuery es una expresión que lee una secuencia de datos XML y devuelve como resultado otra secuencia de datos XML.

La forma típica de una consulta en XQuery es conocida como FLOWR, acrónimo de "For, Let, Where, Order by, Return", que corresponden a las diferentes cláusulas que pueden constituir una consulta[6].

c) XSLT

XSLT es un lenguaje para transformar documentos XML en otros documentos XML[7].

XSLT 1.0 fue publicado como recomendación inmediatamente después del XML 1.0. Probablemente,

archivos HTML a partir de documentos XML.

d) SQL/XML estándar

La ISO/IEC 9075-14:2006 definió una extensión de SQL que le permite trabajar con datos XML, dando origen al estándar SQL/XML. Las ampliaciones definidas se relacionan fundamentalmente a funciones que permiten publicar datos XML, manipular datos XML dentro de la base de datos, y dar soporte al tipo de dato XML.

Líneas de Investigación y desarrollo

Las herramientas investigadas y utilizadas durante el desarrollo del proyecto se detallan a continuación.

Editores XML:

- Altova XMLSpy. Se usó una versión de prueba durante 30 días. Sitio de descarga es <http://www.altova.com>.
- XML Copy Editor: Licencia GPL. Sitio de descarga: <http://xml-copy-editor.sourceforge.net/>.

Sistemas de gestión de bases de datos que manipulan datos XML:

- El SGBD Oracle 10 g con licencia OTN. Sitio de descarga <http://www.oracle.com.ar>
- El SGBD XML nativo eXist 1.4.0, con licencia GPL. Sitio de descarga <http://www.exist-db.org/download.html>

Dentro del proceso de desarrollo del proyecto, además de investigar y profundizar en cada uno de los tópicos significativos de la temática planteada, se realizan ensayos en el tratamiento de datos XML. Concretamente se comenzó una experiencia tomando como escenario la situación presentada en el Departamento de Informática de la Facultad de CEFN de la UNSJ, a raíz del proceso de acreditación de sus carreras. La aplicación de CONEAU necesitaba datos extraídos desde el sistema SIU Guaraní y éste los presentaba en informes o como archivos de extensión txt.

El objetivo de la experiencia fue concretar el uso de las tecnologías investigadas y apreciar sus particularidades y beneficios.

Como el trabajo mencionado no tiene la facultad, ni la finalidad de provocar cambios al SIU, sino simplemente tomar este contexto como ejemplo, la primer tarea efectuada fue crear una base de datos relacional implementada en Oracle 10 g release 2 para "representar" lo que correspondería a la base de datos del SIU. Por supuesto, se tomó un subconjunto muy reducido del modelo completo, a modo de ejemplo. En adelante, a esta base de datos se la referirá como


```

121
122 Select ea.ANIO_ACADEMICO, count(*) Cantidad, da.resultado condicion, ma.NOMBRE
123 From siu.sga_materias ma, siu.sga_atrib_met_plan ca
124 , siu.sga_detalle_acta da, siu.sga_actas_examen ea
125 where ma.departamento='02' and ma.NOMBRE=y and ma.UNIDAD_ACADEMICA=ca.UNIDAD_ACADEMICA and
126 ma.MATERIA=ca.MATERIA and ca.carrera='x' and ca.plan='1990' and ca.version='4' and
127 (da.resultado='R' or da.resultado='A') and da.ACTA=ea.ACTA and da.CARRERA='x' and ea.MATERIA=ma.MATERIA
128 and ea.ANIO_ACADEMICO=y and ea.ANIO_ACADEMICO=z
129 group by ea.ANIO_ACADEMICO, da.resultado, ma.NOMBRE;

```

Figura 5: Consulta en SQL

La Figura 6, presenta el documento resultante al realizar la exportación de la consulta SQL.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE GRID [
<ELEMENT GRID (BANDS, COLUMNS, ROWS)*>
<ATTLIST GRID ExportVersion CDATA #REQUIRED>
<ELEMENT BANDS (BAND)*>
<ELEMENT BAND EMPTY>
<ATTLIST BAND Index CDATA #REQUIRED>
<ATTLIST BAND Caption CDATA #IMPLIED>
<ELEMENT COLUMNS (COLUMN)*>
<ATTLIST COLUMN Name CDATA #REQUIRED>
<ATTLIST COLUMN Caption CDATA #IMPLIED>
<ATTLIST COLUMN Index CDATA #REQUIRED>
<ATTLIST COLUMN BandIndex CDATA #REQUIRED>
<ATTLIST COLUMN RowIndex CDATA #REQUIRED>
<ATTLIST COLUMN ColIndex CDATA #REQUIRED>
<ATTLIST COLUMN Width CDATA #IMPLIED>
<ELEMENT GROUP (GROUP, ROW)*>
<ATTLIST GROUP GroupText CDATA #IMPLIED>
<ELEMENT ROWS (ROW)*>
<ELEMENT ROW EMPTY>
<ATTLIST ROW Column2 CDATA #IMPLIED>
<ATTLIST ROW Column3 CDATA #IMPLIED>
<ATTLIST ROW Column4 CDATA #IMPLIED>
<ATTLIST ROW Column5 CDATA #IMPLIED>
]
<GRID ExportVersion="1.0">
<BANDS>
<BAND Index="0" Caption=""></BAND>
</BANDS>
<COLUMNS>
<COLUMN Name="Column2" Caption="ANIO_ACADEMICO" Index="1" BandIndex="0" RowIndex="0" ColIndex="0" Width="120"></COLUMN>
<COLUMN Name="Column3" Caption="CANTIDAD" Index="2" BandIndex="0" RowIndex="0" ColIndex="1" Width="74"></COLUMN>
<COLUMN Name="Column4" Caption="CONDICION" Index="3" BandIndex="0" RowIndex="0" ColIndex="2" Width="30"></COLUMN>
<COLUMN Name="Column5" Caption="NOMBRE" Index="4" BandIndex="0" RowIndex="0" ColIndex="3" Width="96"></COLUMN>
</COLUMNS>
<ROWS>
<ROW Column2="2001" Column3="48" Column4="A" Column5="Computaci&#243;n F"></ROW>
<ROW Column2="2001" Column3="54" Column4="R" Column5="Computaci&#243;n F"></ROW>
<ROW Column2="2000" Column3="59" Column4="R" Column5="Computaci&#243;n F"></ROW>
<ROW Column2="2002" Column3="59" Column4="R" Column5="Computaci&#243;n F"></ROW>
<ROW Column2="2002" Column3="48" Column4="A" Column5="Computaci&#243;n F"></ROW>
<ROW Column2="2000" Column3="48" Column4="A" Column5="Computaci&#243;n F"></ROW>
</ROWS>
</GRID>

```

Figura 6: Resultado de exportar la consulta SQL

Tal como se puede observar, la exportación genera su DTD dentro del documento.

También es posible exportar el resultado de la consulta asociado con un documento XSL, mostrado en la Figura 7.

```

<?xml version="1.0" ?>
<!-- stylesheets -->
<!-- template match -->
<!-- HTML -->
<!-- BODY -->
<!-- TABLE style="font:10pt MS Shell Dlg 2" border="1" CELSPACING="0" CELLPADDING="1" BGCOLOR="#FFFFFF" -->
<!-- THEAD style="font:bold; background-color:#C0C0C0" -->
<!-- TR valign="top" -->
<!-- csi:for-each select="/GRID/COLUMNS/COLUMN">
<!-- <TD align="left">
<!-- + csi:attribute name="width">
<!-- csi:value-of select="@Caption" />
<!-- </TD>
<!-- </csi:for-each>
<!-- </TR>
<!-- </THEAD -->
<!-- csi:for-each select="GRID/ROWS/ROW">
<!-- <TR valign="top" -->
<!-- <TD align="left">
<!-- <TD align="left">
<!-- csi:value-of select="@Column3" />
<!-- csi:if test="@Column3[.='R']"> </csi:if -->
<!-- <TD -->
<!-- <TD align="left">
<!-- <TD align="left">
<!-- </TR>
<!-- </csi:for-each -->
<!-- </TABLE -->
<!-- </BODY -->
<!-- HTML -->
<!-- template -->
<!-- stylesheets -->

```

Figura 7: Archivo XSL resultante de la exportación

Paso3: Carga de documentos en el Repositorio

El archivo XML generado en el paso anterior, puede ser procesado haciendo uso de cualquiera de los lenguajes existentes para ese propósito. En este caso, se implementó una consulta en XQuery que dió como

el que se cargó en el repositorio.

```

http://localhost:8080/home/siu/xml_consulta_ap_rep_fin_2.xml - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Abre Búsqueda Favoritos
Dirección http://localhost:8080/home/siu/xml_consulta_ap_rep_fin_2.xml
<?xml version="1.0" encoding="UTF-8" ?>
<Resultado_Examen_Materia Nombre_mat="Computación I" Nombre_carrera="Licenciatura en Ciencias de la Información">
<Año_Academico="2000">
<Cantidad>
<Aprobados>48</Aprobados>
<Reprobados>59</Reprobados>
</Cantidad>
</Año>
<Año_Academico="2001">
<Cantidad>
<Aprobados>48</Aprobados>
<Reprobados>54</Reprobados>
</Cantidad>
</Año>
<Año_Academico="2002">
<Cantidad>
<Aprobados>46</Aprobados>
<Reprobados>69</Reprobados>
</Cantidad>
</Año>
</Resultado_Examen_Materia>

```

Figura 8: Documento xml cargado en el repositorio

El objetivo de este paso fue dejar accesible el documento XML para poder ser utilizado por quien lo necesite. Para ello se creó una carpeta dentro del repositorio de Oracle llamada 'siu', haciendo uso del lenguaje PL/SQL. A continuación, en la Figura 9, se muestra el script correspondiente.

```

DECLARE
res BOOLEAN;
BEGIN
res := DBMS_XDB.createFolder('/home/siu/');
END;

```

Figura 9: Creación de la carpeta en el repositorio

Una vez creada la carpeta contenedora, se cargó el documento ejemplo. La Figura 10 muestra el script correspondiente.

```

DECLARE
res BOOLEAN;
BEGIN
res := DBMS_XDB.createResource
('/home/siu/xml_consulta_ap_rep_fin_2.xml',
filename('XMLDIR1', 'xml_consulta_ap_rep_fin_2.xml'),
nls_charset_id('AL32UTF8'));
END;

```

Figura 10: Carga del documento en la carpeta siu

De igual manera se realizó la carga del documento XSL "consulta_ap_rep_final,xsl". Se muestra a continuación en la Figura 11.

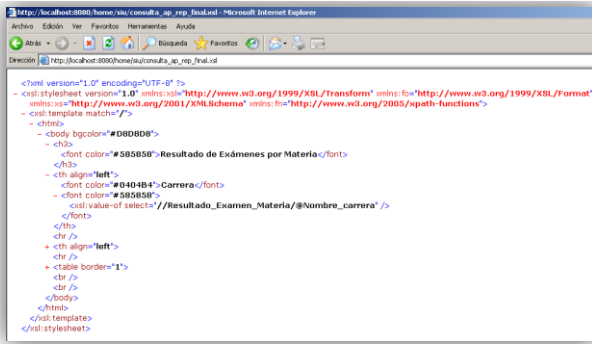


Figura 11: Hoja de estilo cargada al repositorio

La Figura 12 presenta el estado del repositorio luego de la carga de ambos archivos.

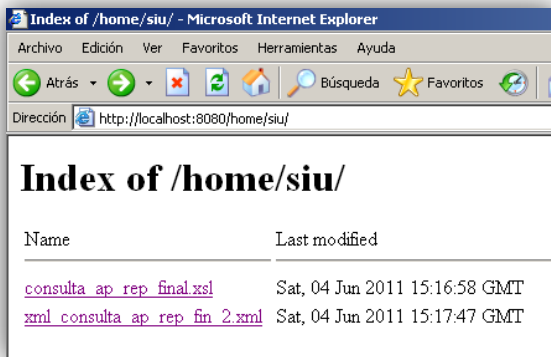


Figura 12: Vista del repositorio desde el navegador

Por último, en la Figura 13, se muestra el documento XML generado al cual se le asoció la plantilla “consulta_ap_rep_final.xml”.

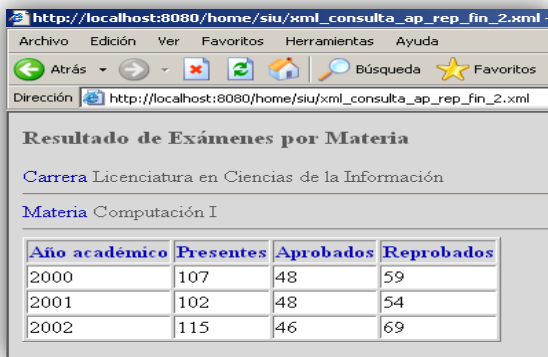


Figura 13: Vista del documento en el navegador

Resultados y Objetivos

Dados los propósitos fundamentales trazados en el proyecto:

- Profundizar en el estudio de la tecnología XML
- Investigar y experimentar con motores de bases de datos que gestionen datos XML.
- Asesorar trabajos finales de grado
- Proponer cursos de actualización
- Elaborar material bibliográfico

Se mencionan los resultados obtenidos hasta el

momento. Este trabajo preliminar y expuesto anteriormente en el punto anterior, nos permitió poner en práctica muchos de los temas estudiados. Sólo se expuso el uso del repositorio de motor Oracle, no obstante, también se hizo el ensayo usando eXist.

- Se dirigió un trabajo final de la carrera Licenciatura en Sistemas de Información, “Tecnología XML y Bases de Datos: Un ejemplo de aplicación con datos del SIU”. C. Vera y L. Romera.
- Se ha comenzado a asesorar otro trabajo final de la carrera LSI, “XML y Bases de datos: Un caso de aplicación”. M. Avendaño.
- Se propuso y acordó el dictado del curso: “Base de Datos Relacional-XML” dirigido a empleados de OSSE (Obras Sanitarias Sociedad del Estado, San Juan). Está planificado en 70 horas presenciales, y se comenzará a dictar a la brevedad.

Formación de Recursos Humanos

Se incorporó una alumna adscripta al proyecto:

- Cristina Vera para trabajar sobre el soporte de XML en distintos motores de bases de datos.

Está planeada la presentación 2 becas de investigación, en la presentación de 2 becas de éste año lectivo, una para alumnos avanzados y la otra de iniciación.

Bibliografía

- [1] Elmasri, Navathe. FUNDAMENTALS OF DATABASE SYSTEMS, Cuarta Edición. Addison Wesley
- [2] Molina, Ullman, Widom. DATABASES SYSTEMS: THE COMPLETE BOOK. Prentice Hall.
- [3] Silberschatz, Korth, Sudarshan. FUNDAMENTOS DE BASES DE DATOS, Sexta Edición, Mc Graw Hill.
- [4] <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>
- [5] W3C XPath. <http://www.w3.org/TR/xpath>.
- [6] W3C XQuery. <http://www.w3.org/TR/XQuery>, <http://www.w3schools.com/xquery/>
- [7] W3C XSL Transformations (XSLT) <http://www.w3.org/TR/xslt.html>