

METODOLOGÍA PARA EVALUAR EL RENDIMIENTO DE SISTEMAS DE ALMACENAMIENTO Y RECUPERACIÓN DE DOCUMENTOS XML EN BASES DATOS NATIVAS PARA XML Y NO NATIVAS HABILITADAS PARA XML.

CLAUDIO ULLOA HEINSOHN

Universidad Austral de Chile, Licenciado en Ciencias de la Ingeniería, caulloah@hotmail.com, Guillermo Lunecke 1075 Valdivia- Chile

JUAN PABLO SALAZAR FERNÁNDEZ

Universidad Austral de Chile, Ingeniero Civil en Informática MBA, juansalazar@uach.cl, General Lagos 2086 Valdivia - Chile

MARTÍN SOLAR MONSALVES

Universidad Austral de Chile, Ingeniero Civil Industrial MBA, msolar@uach.cl, General Lagos 2086 Valdivia - Chile

MIGUELINA VEGA ROSALES

Universidad Austral de Chile, Ingeniero Comercial Magíster en Estadística, mvega@uach.cl., General Lagos 2086 Valdivia - Chile

Resumen - El siguiente proyecto propone una metodología para elegir una base de datos con soporte nativo o habilitado para XML que entregue los mejores rendimientos de almacenamiento, consulta y actualización de documentos XML de acuerdo a los requerimientos de una organización.

En la primera parte se plantean las características de XML y se describe el funcionamiento de Tamino XML Server, Oracle 10g XML DB, dbXML y Xpsql.

En la segunda parte se identifican los parámetros que influyen en la elección de una base de datos y se realizan distintas pruebas que nos permiten formular la metodología.

Abstract - The following project proposes a methodology to choose a database with native support or XML-Enabled that gives the best performance to storage, query and update XML documents according to the organizational requirements.

In the first part, XML characteristics are presented. In addition to the Tamino XML Server, Oracle 10g XML DB, dbXML and Xpsql operation is described.

In the second part parameters that have influence in the election of a database management systems are identified, making different tests that allow us to formulate the methodology.

Palabras Claves – Base de Dato, Metodología, Nativa, Relacional, XML.

1. INTRODUCCIÓN

XML (Extensible Markup Language), se está convirtiendo en el formato de datos de elección para una amplia variedad de soluciones de sistemas de información. Aplicaciones comunes que usan XML incluyen transmisión de documentos en sistemas B2B (Business to Business), construcción de formatos de mensajes para integración de

aplicaciones Internet, almacenamiento y obtención de datos, y varias actividades de manipulación de datos dentro de aplicaciones [1].

La mayoría de las aplicaciones tradicionales de negocio y de las aplicaciones basadas en Internet dependen de bases de datos. Existen diversos modelos diferentes de bases de datos, sin embargo, la mayoría de estos modelos son relacionales [2]. Para mantener datos en una base de datos éstos se deben obtener y almacenar de una manera consistente, confiable y eficiente.

El uso de la infraestructura existente de bases de datos para la administración de demandas de datos XML parece ser lógico. Sin embargo, nuevos productos nativos XML están construidos para manejar las demandas de datos XML en forma nativa sin el bagaje de conversiones a otras estructuras de bases de datos como la relacional. Adicionalmente, existe una diversidad de estrategias de almacenamiento XML, procesos de conversión, y niveles de soporte para XML con los productos líderes de bases de datos [3].

2. OBJETIVOS

2.1. Objetivo General

Desarrollar un modelo que permita escoger el mejor sistema de almacenamiento y manejo de documentos XML en una organización.

2.2. Objetivos Específicos

- Analizar los diferentes productos que permiten el almacenamiento y manejo de documentos XML.

- Definir parámetros que permitan identificar características claves para el almacenamiento y manejo de documentos XML.
- Formular una metodología que permita a una empresa elegir la mejor configuración para el manejo de una base de datos basado en documentos XML.
- Evaluar esta metodología en alguna base de datos de una empresa, institución o a través de un caso práctico.

3. FUNDAMENTOS TEORICOS

XML es un conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes muy simple pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos [4]. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones que además sirve para estructurar, almacenar e intercambiar información.

XML no sólo contiene datos, también puede contener la estructura e incluso la categorización de la información dentro del mismo documento, es decir es autodescriptiva.

En un simple documento XML se pueden identificar entidades o elementos, etiquetas y datos, las etiquetas proporcionan el contexto de cada valor y permiten especificar su semántica.

Los documentos XML según su estructura se pueden agrupar en dos grandes categorías: *centrados en los datos* y *centrados en los documentos*. [3]

Si el documento XML tiene una estructura bien definida y contiene datos actualizables usados en maneras diversas, el documento es típicamente *centrado en los datos*.

Los documentos XML centrados en los datos suelen incluir documentos de facturación típicos, órdenes de compra, y documentos menos estructurados, y es apropiada para contenidos de periódicos, artículos y publicidad.

Los documentos XML *centrados en el documento* tienden a ser más impredecibles en tamaño y contenido que los centrados en los datos los cuales son altamente estructurados, con tipos de datos de tamaño limitado y reglas menos flexibles para campos opcionales y contenido.

Los sistemas de almacenamiento XML se deben acomodar eficientemente con ambos tipos de requerimientos de datos, dado que XML está siendo ampliamente usado en sistemas que administran los dos tipos de datos. La mayoría de los productos se enfocan en servir uno de esos formatos de datos mejor que el otro.

Entre los principales sistemas de almacenamiento de documentos XML se puede distinguir a las Bases de Datos Relacionales habilitadas para XML y las Bases de Datos Nativas para XML (NXDs).

3.1. Bases de Datos Relacionales habilitadas para XML.

Las Bases de Datos Relacionales Habilitadas para XML son Bases de Datos relaciones tradicionales a las cuales se les ha agregado algún soporte para contenido XML. La principal característica agregada es tener un mapeo objeto-relacional entre contenido XML y las tablas en el RDBMS (Relational Data Base Management System). El contenido XML es fragmentado y almacenado en las tablas del RDBMS. El problema de esto es que la fidelidad del DOM (Document Object Model) se pierde durante la fragmentación. También para cierto contenido el mapeo objeto-relacional es muy difícil de lograr. Estas bases de datos manejan tales contenidos realizando este almacenamiento como un CLOB (Character Large Objects). El problema con el tipo de datos CLOB es que si sólo una pequeña porción del documento es modificada, el CLOB completo debería ser leído, modificado y actualizado. Así el rendimiento del modelo de almacenamiento CLOB decrece rápidamente a medida que el tamaño de los documentos almacenados aumenta.

Las bases de datos relacionales Habilitadas para XML son buenas para cierto tipo de contenido de documentos XML centrado en los datos el cual tiene una estructura fija y se ajusta bien a tablas relacionales y no tiene información jerárquica.

Otra ventaja de BD Relacionales Habilitadas para XML es que llevan la madurez de los productos RDBMS con ellos. Las organizaciones han estado almacenando sus datos críticos en RDBMS y han crecido confiando sus datos en ellos.

3.2. Bases de datos nativas para XML (NXDs)

Una NXDs define un modelo lógico para un documento XML y almacena y recupera documentos de acuerdo a este modelo. Como mínimo el modelo debe incluir elementos, atributos, PCDATA (Parsed Character Data) y el orden del documento.

Tiene un documento XML como su unidad fundamental (lógica) de almacenamiento, así como una base de datos relacional tiene una fila en una tabla como su unidad fundamental (lógica) de almacenamiento.

No requiere un particular modelo de capa física de almacenamiento, por ejemplo puede ser construida sobre una base de dato relacional, jerárquica u

orientada a objetos o usar su propio formato de almacenamiento tales como la indexación o la compresión de archivos.

Según esta definición las NXDs no representan un nuevo modelo de bajo nivel de base de datos y no están intentando reemplazar a las base de datos existentes, ellas son simplemente una herramienta prevista para proveer un robusto almacenamiento y manipulación de documentos XML.

Aunque no todas las NXDs son exactamente iguales, existen características básicas similares que se aplican a la mayoría de los productos actuales:

- Almacenamiento

Existen varias razones para almacenar documentos XML en una NXD, la primera de estas es cuando los datos son semi-estructurados y si se desean mapear a una base de datos relacional resulta en un gran número de columnas con valores null o gran cantidad de tablas, lo cual es ineficiente..

Otra razón para almacenar información en una NXDs es la velocidad para recuperar datos. Dependiendo de cómo almacena los datos físicamente una NXD puede ser capaz de recuperar datos mucho más rápido que una base de datos relacional, esto es porque algunas de las estrategias de almacenamiento usadas por las NXD permiten almacenar físicamente documentos XML enteros o usar punteros físicos entre las partes del documento.

- Colecciones de Documentos XML

Muchas NXDs soportan el concepto de colección (collection), esto juega un papel similar al de una tabla en una base de datos relacional

- Lenguajes de Consulta

Casi todas las NXDs soportan uno o más lenguajes de consulta. Los más populares de estos lenguajes son XPath (XML Path Language) y XQuery (XML Query), aunque existen numerosos lenguajes de consulta propietarios que son bien soportados.

- Actualizar y Borrar

Las NXDs tienen una variedad de estrategias para actualizar y borrar documentos, desde simplemente reemplazando o borrando el documento existente hasta lenguajes que especifican cómo modificar fragmentos de un documento. La mayoría de estos métodos son propietarios, sin embargo existen dos lenguajes estándar para actualizar documentos XML: XUpdate (XML Update) y un conjunto de extensiones de XQuery

- Indexación

Todas las NXDs soportan indexación como una manera de incrementar la velocidad de las consultas.

4. PRODUCTOS

4.1. Oracle 10g XML DB

Oracle 10g XML DB extiende la Base de Datos, entregando la funcionalidad asociada tanto con NXD como una BD relacional, se incluyen las siguientes características:

Provee métodos de acceso estándar para la navegación y consulta de documentos XML, absorbiendo estos modelos dentro de la BD Oracle 10g.

Permite almacenar, consultar, actualizar, transformar o procesar documentos XML, mientras que al mismo tiempo provee acceso SQL a los mismos datos XML. De igual manera, permite operaciones XML sobre datos SQL.

Incluye un repositorio XML que permite al contenido XML ser organizado y administrado usando una metáfora Archivo/Carpeta/URL

Provee una infraestructura de almacenamiento, contenido y lenguaje de programación independiente para almacenar y administrar datos XML.

Provee métodos estándares para acceder y actualizar documentos XML tales como FTP, HTTP y WebDAV. Permite el uso de APIs estándar para acceder y manipular el contenido XML usando Java, C y PL/SQL.

Oracle XML DB no es un servidor separado, el nombre es para distinguir un grupo de tecnologías relacionadas al desempeño de almacenamiento y recuperación de documentos XML disponibles en la Base de Datos Oracle.

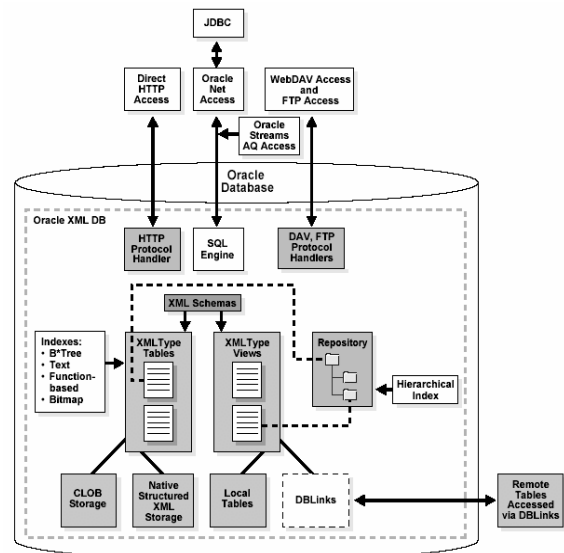


Fig.1. Arquitectura Oracle 10g XML DB.

El corazón de Oracle XML DB es XMLType, un tipo de datos que soporta comprensivamente un conjunto de métodos contruidos para el procesamiento de los documentos XML a través de su ciclo de vida. XMLType puede representar un documento XML como una instancia (de XMLType) en SQL. Las instancias de XMLType pueden ser almacenadas usando un almacenamiento tanto estructurado como no estructurado.

En la Fig. 1, cuando el esquema XML es registrado en Oracle XML DB, los elementos XML para las tablas XMLType, las tablas con columnas XMLType y las vistas XMLType son mapeadas a las tablas de la base de datos. Esto se puede ver y acceder en el repositorio XML.

Los datos en las tablas XMLType y las tablas con columnas XMLType pueden ser almacenados como un CLOB o nativamente usando una estructura XML.

Los datos en las vistas de los XMLType pueden ser almacenadas en tablas locales o tablas remotas, las que se pueden acceder a través de DBlinks. Tanto tablas y vistas XMLType pueden ser indexados.

Existen varias opciones para acceder a los datos en el repositorio XML: HTTP, WebDAV y FTP ySQL a través de Oracle Net Services incluyendo JDBC.

XPath reescrito es la clave para mejorar el desempeño de las sentencias SQL que contienen expresiones XPath convirtiendo las funciones en convencionales sentencias SQL relacionales. Cuando los datos son almacenados en XMLType (objeto –relacional) usando un esquema XML y se realizan consultas con XPath, estas consultas pueden ser potencialmente rescritas directamente a la subcapa de las columnas objeto-relacionales.

4.2. Tamino XML Server.

Tamino XML Server es una plataforma para la administración de datos y se basa en XML y otros estándares de tecnologías de Internet. Como característica principal se puede señalar el almacenamiento de documentos XML en forma nativa. El corazón de Tamino XML Server es un servidor base que provee un mecanismo de alto desempeño y funcionalidad.

Tamino habilita servicios que proveen un amplio rango de herramientas y componentes necesarios para el desarrollo productivo de soluciones basadas en XML enfocadas al Web, como la publicación e intercambio de documentos electrónicos sobre Internet.

La Fig. 2 describe cómo la tecnología de Tamino XML Server es implementada. En esencia Tamino

consiste de dos partes principales, el Tamino XML Server y los componentes productos (servicios habilitados), los cuales son simples unidades capaces de trabajar como componentes aislados

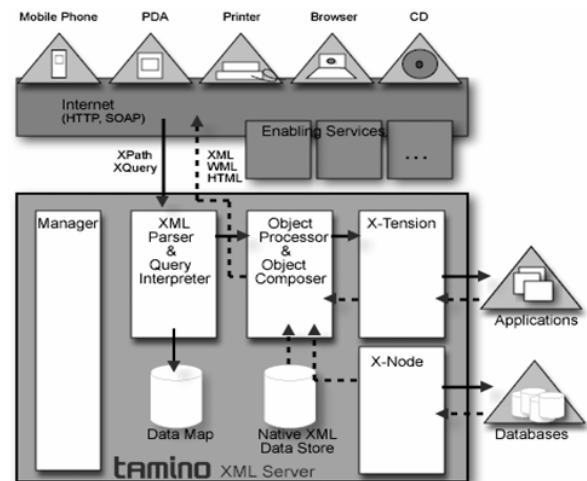


Fig.2. Arquitectura Tamino XML Server

Existen varios componentes Tamino que permiten llevar a cabo la realización de tareas de forma gráfica, los principales componentes son:

- Editor de Esquemas Tamino

El Editor de Esquemas Tamino permite generar un esquema Tamino que es un archivo de acuerdo a los estándares de esquemas XML, es posible también importar DTDs y esquemas SQL y convertirlos en esquemas Tamino.

- XML Loaders

Tamino provee herramientas para cargar datos XML dentro de la base de datos desde una fuente externa.

- X-Plorer

X-Plorer es una herramienta para consulta, exploración y manipulación de los contenidos del Tamino XML Server.

- XQuery Tool

XQuery Tool es una herramienta para consulta y manipulación (actualización) de los contenidos del Tamino XML server.

- Tamino Interactive Interface

El Tamino Interactive Interface es una interfaz basada en el explorador que permite definir y borrar colecciones y esquemas, cargar y borrar instancias XML de un esquema y consultar a una base de datos.

- Almacenamiento

Para realizar el almacenamiento de datos XML en Tamino lo primero es definir un esquema de acuerdo a los estándares que conforman un esquema XML,

luego se cargan los datos de acuerdo a este esquema a través de la interfaz interactiva o usando el cargador de datos Tamino.

▪ Lenguajes de Consulta

Tamino soporta dos lenguajes de consulta, el Tamino X-Query (basado en el estándar XPath) y el lenguaje de consulta estándar XQuery recomendado por el W3C Tamino.

4.3. dbXML.

dbXML es una base de datos XML Nativa, desarrollada usando el Java 2 Standard Edition versión 1.4 y puede operar apropiadamente en todas las plataformas en las cuales J2SE 1.4 ha sido portada.

dbXML maneja documentos en colecciones las que pueden ser vista como una estructura jerárquica, en forma similar a como trabaja la estructura de directorios de un sistema operativo. Una simple colección puede ser asociada con múltiples indexaciones, extensiones, triggers y colecciones de hijos.

El motor de almacenaje de una colección es llamado un filer. Por defecto dbXML usa un filer nativo llamado BTreeFiler, pero otros filers son disponibles para propósitos específicos, los que se muestran transparentes a los usuarios.

Aunque los documentos en una colección no necesitan ser vinculados a un esquema común, es una buena práctica hacer una simple colección que contenga documentos similares para una fácil indexación y consulta sobre esta colección.

dbXML provee varios sistemas de consulta y actualización a los documentos XML o parte de ellos, entre los que se encuentran XPath, XSLT, XUpdate y FullText.

Las extensiones son una manera de agregar capacidades extra al servidor dbXML. Estas extensiones son clases de Java que implementan una interfaz y cuyos métodos públicos se muestran como un servicio Web.

dbXML provee capacidades transaccionales a nivel de API, a este nivel todas las acciones de la base de datos pueden ser ejecutadas usando un referencia transaccional y sólo son permitidas las que son seguras.

4.4. XpSQL.

XpSQL es una base de dato multifuncional, escrito completamente en C y construida sobre PostgreSQL que almacena los documentos XML descomponiéndolos en fragmentos y almacenando

estos fragmentos en un conjunto de tablas predefinidas. Para el usuario es transparente la manera como estos fragmentos son almacenados.

XpSQL posee una utilidad a través de la línea de comandos (pgxload) para cargar documentos XML en la base de datos, así como funciones de PostgreSQL para recuperar fragmentos de documentos para cada nodo (identificado por ID). También posee funciones de PostgreSQL que implementan DOM nivel 2 y XPath.

XpSQL utiliza XPath para realizar consultas a los documentos XML almacenados en la base de datos. Existen dos funciones principales de XPath.

5. ANALISIS

Después de analizar la información relacionada al funcionamiento de las Bases de Datos en estudio, se determinaron los parámetros más relevantes que permitan tomar una mejor decisión de cuál Base de Datos se debe utilizar para el manejo de documentos XML, los cuales se detallan a continuación

5.1. Según Orientación de los Datos.

Según la orientación de los datos, los documentos XML se pueden clasificar en dos categorías, Centrados en los Documentos o Centrados en los Datos.

5.2. Almacenamiento.

El almacenamiento es un parámetro importante, ya que por ejemplo mucha de la información XML sólo es almacenada para luego ser procesada por otra aplicación, en este aspecto se debe considerar el uso de espacio (Datos/ Espacio Total), tamaño máximo y tiempo de carga.

5.3. Consulta.

Los tiempos de respuesta de la consulta sobre un documento XML van a depender de cómo esté almacenada la información y si se desea ver parte de un documento XML, un documento completo o una colección de estos (complejidad de la consulta). Para esto se debe tener en cuenta el volumen (cantidad de documentos, porcentaje parte/todo), frecuencia y complejidad de las consultas.

5.4. Actualización.

La actualización de datos es un proceso importante, en los documentos XML no es la excepción y su tiempo de respuesta va a depender del volumen (tamaño de documentos, cantidad de documentos y porcentaje parte/ todo), frecuencia y si es sobre los datos o la estructura.

5.5. Borrado.

Este parámetro también influirá en la elección de una base de datos, por ello se debe considerar el borrado de partes de un documento XML o de todo un documento XML.

5.6. Tipo almacenamiento según aplicación.

El tipo de almacenamiento considera cómo serán almacenados los datos internamente dentro de la base de datos, se pueden distinguir tres tipos:

- CLOB
- Mapeo XML dentro de un Esquema Relacional
- Almacenamiento XML en Forma Nativa.

6. PRUEBAS

En esta etapa se desarrollarán las pruebas de almacenamiento, consulta y actualización de datos XML sobre las cuatro Bases de Datos en estudio con el objetivo de determinar comportamientos de los datos para poder elaborar una metodología que permita escoger la base de datos más adecuada desde un punto de vista técnico.

Cada prueba se repite 10 veces descartando los valores extremos y se trabaja con el tiempo de respuesta promedio esto se debe a que el coeficiente de variación (CV) tiende a cero para todas las configuraciones en las distintas pruebas a realizar.

Las pruebas de borrado se omiten ya que un borrado de datos (no de estructura) es en si una actualización de los datos en blanco, además las actualizaciones serán sobre los datos no sobre la estructura omitiendo este tipo de pruebas.

Los documentos XML corresponden a la descripción de un currículum, el cual tendrá distintas configuraciones de acuerdo a la complejidad de la estructura y a la cantidad de archivos. Según la complejidad tendremos documentos XML con 10, 50, 100 o 500 elementos y según la cantidad de archivos existirían colecciones de documentos XML de 10, 100, 800, 1000 o 5000 archivos

En la Fig. 3 se muestra el esquema general de un currículum del cual se generarán los diferentes niveles de complejidad de un documento XML.

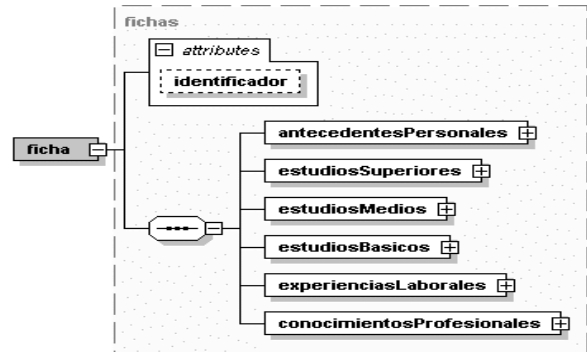


Fig.3. Esquema general de un currículum

Tamino XML Server y Oracle 10g XMLDB serán usados sobre Windows (Windows 2000 Advanced Server) y dbXML y XpSQL sobre Linux (Fedora Core 3).

Por último las pruebas se realizarán sobre un PC con las siguientes características: procesador Pentium IV de 2.26 GHz, disco duro de 80 GB de 7200 RPM, 640 MB de memoria RAM DDR de 333 MHz.

6.1. Almacenamiento.

El proceso de almacenamiento de documentos XML varía de acuerdo a la base de datos a usar, a continuación se muestran los tiempos de respuesta en segundos de cada base de datos sobre todos los tipos de configuración.

La Tabla 1 muestra los tiempos para Tamino.

Tabla 1. Tiempos de almacenamiento en Tamino

		Nº de Elementos			
		10	50	100	500
Nº de Archivos	10	8.417	9.238	9.463	14.103
	100	8.853	9.568	10.143	15.600
	800	11.174	13.348	16.455	50.121
	1000	12.198	15.110	20.063	No
	5000	29.279	46.506	69.681	Aplicable

Cabe señalar que Oracle 10g XML DB presenta dos tipos de almacenamientos, como un CLOB según Tabla 2 y con un mapeo objeto relacional según Tabla 3.

Tabla 2. Tiempos de almacenamiento en Oracle CLOB

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	1.390	1.394	1.410	3.117
	100	2.172	2.280	2.492	6.236
	800	8.720	9.321	11.250	28.678
	1000	9.670	10.972	13.165	34.235
	5000	46.460	57.884	73.965	225.144

Tabla 3. Tiempos de almacenamiento en Oracle Mapeo

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.236	0.292	0.320	0.378
	100	1.431	1.609	1.653	1.793
	800	8.316	9.820	10.620	12.384
	1000	9.532	11.406	12.840	15.010
	5000	62.217	91.085	104.325	134.682

La Tabla 4 y la Tabla 5 muestran los tiempos de almacenamiento para dbXML y XpSQL respectivamente.

Tabla 4. Tiempos de almacenamiento en dbXML

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.366	0.374	0.404	0.620
	100	1.918	2.015	2.190	4.930
	800	10.316	11.720	14.825	37.540
	1000	12.122	15.135	17.115	47.012
	5000	65.654	70.193	85.691	228.644

Tabla 5. Tiempos de almacenamiento en XpSQL

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	14.367	15.128	16.433	25.833
	100	20.350	21.359	23.112	37.616
	800	35.252	39.931	44.856	109.178
	1000	40.612	44.847	52.231	137.812
	5000	125.944	148.771	168.336	547.321

6.2. Consultas.

Para las cuatro bases de datos se realizaron tres tipos de consulta:

Query1: Recuperar un documento completo cuyo identificador es igual a 4. La Tabla 6, Tabla 7, Tabla 8, Tabla 9 y Tabla 10 muestran los tiempos de respuesta para Tamino, Oracle CLOB, Oracle Mapeo, dbXML y XpSQL respectivamente

Tabla 6. Tiempos de Query1 en Tamino

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.005	0.006	0.009	0.029
	100	0.018	0.023	0.032	0.106
	800	0.156	0.210	0.301	1.130
	1000	0.176	0.263	0.410	No Aplicable
	5000	0.991	1.500	2.443	

Tabla 7. Tiempos de Query1 en Oracle CLOB

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.303	1.015	1.831	9.229
	100	0.476	1.195	2.066	10.534
	800	1.782	2.652	3.703	15.298
	1000	2.225	3.123	4.147	18.024
	5000	9.972	12.061	14.024	51.449

Tabla 8. Tiempos de Query1 en Oracle Mapeo

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.069	1.280	1.820	3.280
	100	0.069	1.310	1.840	3.297
	800	0.074	1.400	1.870	3.433
	1000	0.076	1.490	1.960	3.492
	5000	0.601	2.520	3.120	4.980

Tabla 9. Tiempos de Query1 en dbXML

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.038	0.037	0.034	0.085
	100	0.207	0.210	0.226	0.679
	800	0.947	1.302	1.696	5.236
	1000	1.184	1.625	2.121	7.303
	5000	5.697	6.321	8.617	33.684

Tabla 10. Tiempos de Query1 en XpSQL

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.221	0.920	1.645	8.016
	100	0.396	1.019	1.895	9.413
	800	1.703	2.521	3.412	12.781
	1000	2.019	2.936	3.875	14.753
	5000	9.565	10.974	12.323	39.622

Query2: Recuperar parte de un documento (antecedentes personales) cuyo identificador es igual a 120. La Tabla 11, Tabla 12, Tabla 13, Tabla 14 y Tabla 15 muestran los tiempos de respuesta para Tamino, Oracle CLOB, Oracle Mapeo, dbXML y XpSQL respectivamente.

Tabla 11. Tiempos de Query2 en Tamino

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.005	0.006	0.008	0.027
	100	0.021	0.027	0.035	0.110
	800	0.153	0.212	0.313	1.140
	1000	0.191	0.288	0.440	No Aplicable
	5000	1.096	1.560	2.471	

Tabla 12. Tiempos de Query2 en Oracle CLOB

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.352	0.357	0.385	0.575
	100	0.523	0.553	0.586	1.041
	800	1.823	2.045	2.210	5.418
	1000	2.287	2.556	2.762	7.023
	5000	10.109	11.301	12.458	40.480

Tabla 13. Tiempos de Query2 en Oracle Mapeo

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.047	0.055	0.059	0.066
	100	0.047	0.061	0.065	0.077
	800	0.052	0.063	0.069	0.084
	1000	0.052	0.066	0.074	0.089
	5000	0.098	0.257	0.377	0.554

Tabla 14. Tiempos de Query2 en dbXML

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.026	0.027	0.031	0.076
	100	0.126	0.170	0.224	0.668
	800	0.943	1.291	1.694	5.150
	1000	1.179	1.606	2.117	6.523
	5000	5.680	8.116	10.643	32.258

Tabla 15. Tiempos de Query2 en XpsQL

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.240	0.252	0.294	0.599
	100	0.418	0.438	0.477	0.965
	800	1.703	1.897	1.990	5.296
	1000	2.092	2.283	2.482	7.347
	5000	9.697	9.724	10.827	36.637

Query3: Recuperar parte de varios documentos (antecedentes personales) cuya ciudad de origen sea Valdivia. La Tabla 16, Tabla 17, Tabla 18, Tabla 19 y Tabla 20 muestran los tiempos de respuesta para Tamino, Oracle CLOB, Oracle Mapeo, dbXML y XpsQL respectivamente.

Tabla 16. Tiempos de Query3 en Tamino

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.005	0.006	0.008	0.027
	100	0.021	0.027	0.035	0.110
	800	0.153	0.212	0.313	1.140
	1000	0.191	0.288	0.440	No Aplicable
	5000	1.096	1.560	2.471	

Tabla 17. Tiempos de Query3 en Oracle CLOB

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.636	0.652	0.695	1.057
	100	0.813	0.833	0.886	1.419
	800	2.108	2.373	2.508	5.668
	1000	2.572	2.886	3.053	6.620
	5000	10.328	11.660	12.679	39.572

Tabla 18. Tiempos de Query3 en Oracle Mapeo

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.095	0.112	0.120	0.134
	100	0.102	0.118	0.124	0.136
	800	0.109	0.134	0.142	0.167
	1000	0.110	0.136	0.147	0.175
	5000	0.115	0.246	0.308	0.424

Tabla 19. Tiempos de Query3 en dbXML

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.025	0.029	0.033	0.082
	100	0.132	0.177	0.240	0.704
	800	0.922	1.392	1.789	5.489
	1000	1.152	1.740	2.237	6.553
	5000	5.844	8.528	11.311	33.391

Tabla 20. Tiempos de Query3 en XpsQL

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.250	0.320	0.460	2.410
	100	0.512	0.610	0.812	3.200
	800	1.812	2.198	2.567	7.322
	1000	2.115	2.376	2.633	8.590
	5000	9.891	10.119	11.853	37.839

6.3. Actualización.

Se realizaron tres tipos de procesos de actualización en cada base de datos:

Update1: Actualizar un documento completo cuyo identificador es igual a 4. La Tabla 21, Tabla 22, Tabla 23, Tabla 24 y Tabla 25 muestran los tiempos de respuesta para Tamino, Oracle CLOB, Oracle Mapeo, dbXML y XpsQL respectivamente.

Tabla 21. Tiempos de Update1 en Tamino

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.011	0.012	0.014	0.032
	100	0.030	0.045	0.055	0.189
	800	0.155	0.240	0.300	1.935
	1000	0.242	0.300	0.390	No Aplicable
	5000	1.325	1.550	2.039	

Tabla 22. Tiempos de Update1 en Oracle CLOB

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.016	0.016	0.018	0.062
	100	0.128	0.132	0.158	0.591
	800	1.033	0.943	1.333	4.361
	1000	1.234	1.190	1.591	5.249
	5000	6.155	7.583	8.499	33.991

Tabla 23. Tiempos de Update1 en Oracle Mapeo

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.016	0.016	0.017	0.021
	100	0.016	0.017	0.018	0.027
	800	0.017	0.019	0.033	0.100
	1000	0.017	0.022	0.039	0.132
	5000	0.021	0.104	0.200	0.845

Tabla 24. Tiempos de Update1 en dbXML

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.429	0.461	0.547	1.423
	100	1.633	1.996	2.418	7.061
	800	11.193	13.856	16.904	51.872
	1000	13.991	17.320	21.130	66.299
	5000	66.748	88.526	119.933	331.496

Tabla 25. Tiempos de Update1 en XpSQL

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.629	0.721	0.862	1.633
	100	1.841	2.321	2.872	7.054
	800	13.125	14.618	19.667	55.321
	1000	15.258	19.054	23.116	68.225
	5000	69.123	91.656	123.112	342.186

Update2: Actualizar parte de un documento (antecedentes personales) cuyo identificador es igual a 120. La Tabla 26 Tabla 27, Tabla 28, Tabla 29 y Tabla 30 muestran los tiempos de respuesta para Tamino, Oracle CLOB, Oracle Mapeo, dbXML y XpsQL respectivamente.

Tabla 26. Tiempos de Update2 en Tamino

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.011	0.013	0.016	0.033
	100	0.029	0.042	0.056	0.188
	800	0.184	0.240	0.340	1.932
	1000	0.242	0.290	0.410	No Aplicable
	5000	1.327	1.590	2.038	

Tabla 27. Tiempos de Update2 en Oracle CLOB

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.019	0.022	0.030	0.102
	100	0.138	0.147	0.171	0.605
	800	1.042	1.086	1.377	4.361
	1000	1.248	1.430	1.588	6.663
	5000	6.187	7.520	8.535	33.475

Tabla 28. Tiempos de Update2 en Oracle Mapeo

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.016	0.016	0.017	0.019
	100	0.028	0.029	0.031	0.045
	800	0.031	0.043	0.047	0.140
	1000	0.033	0.050	0.052	0.182
	5000	0.145	0.218	0.296	1.090

Tabla 29. Tiempos de Update2 en dbXML

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.433	0.477	0.571	1.535
	100	1.639	2.028	2.539	7.640
	800	10.826	13.678	16.245	52.363
	1000	14.074	17.886	21.985	59.128
	5000	67.001	89.150	120.207	332.789

Tabla 30. Tiempos de Update2 en XpSQL

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.635	0.965	1.214	3.087
	100	1.853	2.456	4.046	14.002
	800	13.347	17.521	22.318	58.833
	1000	15.658	21.996	29.998	72.113
	5000	71.322	95.184	124.114	344.277

Update3: Actualizar parte de varios documentos (antecedentes personales) cuya ciudad de origen sea Valdivia. La Tabla 31, Tabla32, Tabla 33, Tabla 34 y Tabla 35 muestran los tiempos de respuesta para Tamino, Oracle CLOB, Oracle Mapeo, dbXML y XpsQL respectivamente

Tabla 31. Tiempos de Update3 en Tamino

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.016	0.016	0.019	0.044
	100	0.030	0.044	0.061	0.198
	800	0.201	0.260	0.320	1.963
	1000	0.233	0.300	0.420	No Aplicable
	5000	1.259	1.560	2.001	

Tabla 32. Tiempos de Update3 en Oracle CLOB

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.041	0.041	0.045	0.134
	100	0.212	0.224	0.241	0.733
	800	1.663	1.963	2.021	5.044
	1000	1.911	2.099	2.270	6.002
	5000	9.500	10.930	11.859	38.767

Tabla 33. Tiempos de Update3 en Oracle Mapeo

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.018	0.022	0.024	0.027
	100	0.031	0.044	0.048	0.062
	800	0.040	0.102	0.123	0.184
	1000	0.041	0.111	0.139	0.239
	5000	0.052	0.483	0.720	1.219

Tabla 34. Tiempos de Update3 en dbXML

		N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.435	0.519	0.628	1.551
	100	1.659	2.193	2.759	7.682
	800	12.411	15.231	18.633	54.869
	1000	14.743	18.940	24.130	62.148
	5000	68.358	93.458	122.318	337.476

Tabla 35. Tiempos de Update3 en XpSQL

	N° de Elementos	N° de Elementos			
		10	50	100	500
N° de Archivos	10	0.639	1.063	1.409	4.100
	100	1.921	2.741	3.864	15.337
	800	13.866	17.612	23.617	60.512
	1000	16.023	21.322	28.326	73.654
	5000	72.145	98.482	128.471	356.133

7. ANÁLISIS DE RESULTADOS

Este análisis muestra cómo se comportan las bases de datos en estudio en las operaciones de almacenamiento, consulta y actualización de acuerdo a la complejidad de la estructura (expresada en la cantidad de elementos de ésta) y la cantidad de archivos de los documentos XML.

Es importante considerar que una tendencia lineal presenta un acceso secuencial a los datos con recorridos de orden N, una tendencia polinomial de grado 2 realiza un acceso secuencial con recorridos del orden N x M, por su parte una tendencia logarítmica se relaciona con un acceso indexado.

Tanto para el almacenamiento, como para los distintos niveles de complejidad de consultas y actualizaciones, el tiempo de respuesta de Tamino presenta una tendencia Polinómica de grado 2 creciente y positiva con tiempos de respuesta muy buenos.

El tiempo de respuesta en Oracle CLOB presenta una tendencia Polinómica de grado 2 creciente y positiva en el almacenamiento y la actualización en sus distintos niveles de complejidad tanto para la estructura como para la cantidad de archivos. Las consultas muestran esta tendencia solamente para la estructura., para la cantidad de archivos presenta una tendencia lineal.

El tiempo de respuesta en Oracle Mapeo presenta una tendencia Polinómica de grado 2 crecientes y positiva para almacenamiento, consulta y actualización para la cantidad de archivos y una tendencia logarítmica en el almacenamiento, consulta y para una actualización compleja según la estructura. En el caso de la actualización para estructuras simples y medianas presenta una tendencia Lineal.

dbXML según el número de archivos, el tiempo de respuesta presenta una tendencia Lineal para el almacenamiento, consultas y actualización y según la estructura una tendencia Polinómica de grado 2 creciente y positiva para el almacenamiento, consultas (simple y media) y actualización y una tendencia Lineal para la consulta compleja.

XpSQL según el número de archivos, el tiempo de respuesta presenta una tendencia Lineal para el

almacenamiento, consultas y actualización y según la estructura una tendencia Polinómica de grado 2 creciente y positiva para el almacenamiento y consultas y una tendencia Lineal para la actualización.

Los tipos de tendencias presentes tanto para la complejidad de la estructura como para la cantidad de archivos se resumen en las figuras Fig. 4, Fig. 5, Fig.6, Fig7 y Fig. 8.

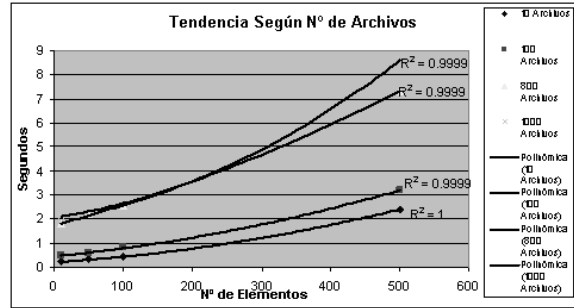


Fig.4. Tendencia Polinomial de grado 2 según N° de archivos

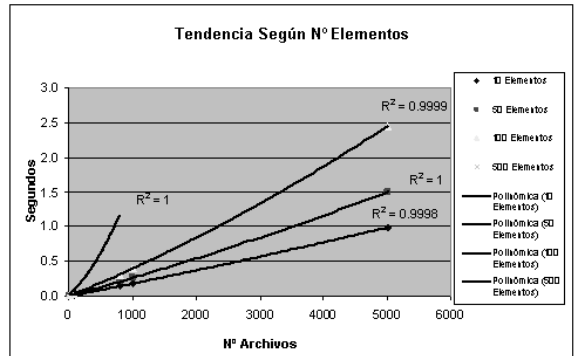


Fig.5. Tendencia Polinomial de grado 2 según N° de elementos

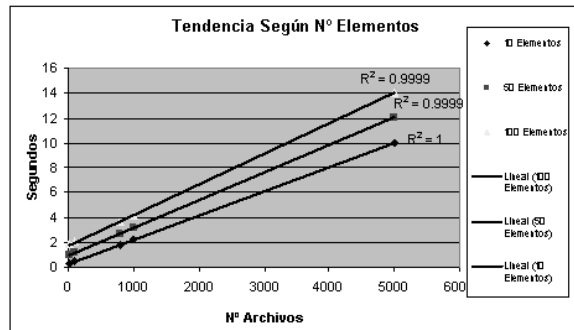


Fig.6. Tendencia Lineal según N° de elementos

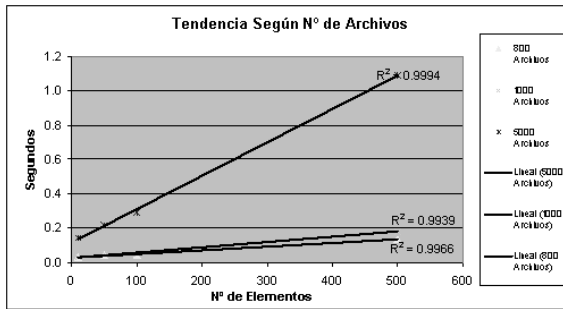


Fig.7. Tendencia Lineal según N° de archivos

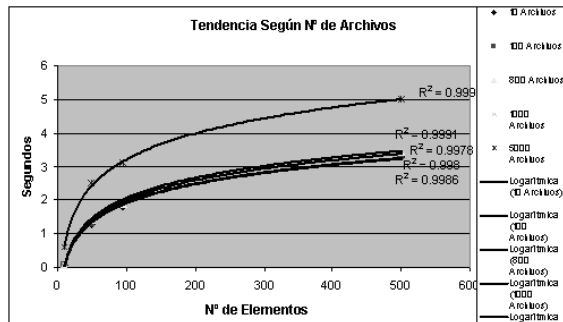


Fig.8. Tendencia Logarítmica según N° archivos

Según los tiempos de respuesta de las cuatro bases de datos en estudio, tanto para el almacenamiento, consultas y actualización podemos agrupar en Mejor rendimiento, rendimiento Bueno o Bajo rendimiento de acuerdo a la cantidad de archivos a almacenar y a la complejidad de la estructura asignándoles un peso de 5, 3 y 1 respectivamente.

El Almacenamiento muestra que a baja cantidad de archivos los productos muestran un buen rendimiento salvo XpSQL que presenta el peor rendimiento en todos los tipos de configuración. Cuando la cantidad de archivos aumenta Tamino presenta mejores tiempos de respuesta. La complejidad de la estructura afecta más a Oracle Mapeo OR con mejores tiempos de respuesta para una estructura más compleja, esto se muestra en la Tabla 36.

Los tiempos de respuesta para las consultas según la cantidad de archivos muestran a Tamino y Oracle Mapeo OR con los mejores rendimientos, dbXML presenta buenos tiempos de respuesta mientras que Xpsql y Oracle CLOB presentan malos tiempos de respuesta. Para estructuras complejas, Tamino y Oracle Mapeo OR presentan mejores rendimientos y según la complejidad de la consulta estos rendimientos se mantienen.

Tabla 36. Rendimientos Almacenamiento

Cantidad de archivos	Nivel	Estructura		
		Simple (10 Elementos)	Mediana (50 -100 Elementos)	Compleja (500 Elementos)
Baja (10 -100 Archivos)	Mejor (5)	Oracle Clob	Oracle Mapeo OR	Oracle Mapeo OR
		Oracle Mapeo OR	Oracle Clob	Oracle Clob
		dbXML	dbXML	dbXML
	Bueno (3)	Tamino	Tamino	Tamino
		XpSQL	XpSQL	XpSQL
		Oracle Clob	Oracle Clob	XpSQL
Media (800 -1000 Archivos)	Mejor (5)	Oracle Mapeo OR	Oracle Mapeo OR	Oracle Mapeo OR
		Tamino	dbXML	Oracle Clob
		dbXML	Tamino	dbXML
	Bueno (3)	XpSQL	XpSQL	XpSQL
		Tamino	Tamino	Oracle Mapeo OR
		Oracle Clob	Oracle Clob	Tamino
Alta (5000 Archivos)	Mejor (5)	Oracle Mapeo OR	dbXML	Oracle Clob
		Oracle Mapeo OR	dbXML	dbXML
		dbXML	XpSQL	XpSQL
	Bueno (3)	XpSQL	XpSQL	XpSQL
		Oracle Clob	Oracle Clob	Oracle Mapeo OR
		Oracle Mapeo OR	dbXML	Oracle Clob
Malo (1)	dbXML	Oracle Mapeo OR	dbXML	
	XpSQL	XpSQL	XpSQL	
	Oracle Clob	Oracle Clob	Oracle Clob	

En la Tabla 37 y Tabla 38 se presentan los rendimientos de Query1, Query2 y Query3, los dos últimos agrupados debido a que sus rendimientos son los mismos.

Tabla 37. Rendimientos Query1

Cantidad de archivos	Nivel	Estructura		
		Simple (10 Elementos)	Mediana (50 -100 Elementos)	Compleja (500 Elementos)
Baja (10 -100 Archivos)	Mejor (5)	Tamino	Tamino	Tamino
		Oracle Mapeo OR	dbXML	dbXML
		Oracle Mapeo OR	Oracle Mapeo OR	Oracle Mapeo OR
		dbXML	XpSQL	XpSQL
		XpSQL	Oracle Clob	Oracle Clob
		Oracle Clob	Oracle Clob	Oracle Clob
	Bueno (3)	Oracle Mapeo OR	Tamino	Tamino
		Tamino	Oracle Mapeo OR	Oracle Mapeo OR
		dbXML	dbXML	dbXML
		XpSQL	XpSQL	XpSQL
		Oracle Clob	Oracle Clob	Oracle Clob
		Oracle Mapeo OR	Tamino	Oracle Mapeo OR
Alta (5000 Archivos)	Mejor (5)	Tamino	Oracle Mapeo OR	Tamino
		Oracle Mapeo OR	dbXML	dbXML
		dbXML	XpSQL	XpSQL
	Bueno (3)	XpSQL	XpSQL	XpSQL
		Oracle Clob	Oracle Clob	Oracle Clob
		Oracle Mapeo OR	Tamino	Oracle Mapeo OR
Malo (1)	Oracle Clob	Oracle Clob	Oracle Clob	
	XpSQL	XpSQL	XpSQL	
	Oracle Clob	Oracle Clob	Oracle Clob	

Tabla 38. Rendimientos Query2 y Query3

Cantidad de archivos	Nivel	Estructura		
		Simple (10 Elementos)	Mediana (50 -100 Elementos)	Compleja (500 Elementos)
Baja (10 -100 Archivos)	Mejor (5)	Oracle Mapeo OR	Oracle Mapeo OR	Oracle Mapeo OR
		Tamino	Tamino	Tamino
		dbXML	dbXML	dbXML
		XpSQL	XpSQL	XpSQL
		Oracle Clob	Oracle Clob	Oracle Clob
		Oracle Mapeo OR	Oracle Mapeo OR	Oracle Mapeo OR
	Bueno (3)	Tamino	Tamino	Tamino
		dbXML	dbXML	dbXML
		XpSQL	XpSQL	XpSQL
		Oracle Clob	Oracle Clob	Oracle Clob
		Oracle Mapeo OR	Oracle Mapeo OR	Oracle Mapeo OR
		Tamino	Tamino	Tamino
Alta (5000 Archivos)	Mejor (5)	Oracle Mapeo OR	Oracle Mapeo OR	Oracle Mapeo OR
		Tamino	dbXML	dbXML
		XpSQL	XpSQL	XpSQL
	Bueno (3)	Oracle Clob	Oracle Clob	Oracle Clob
		Oracle Mapeo OR	Oracle Mapeo OR	Oracle Mapeo OR
		Tamino	dbXML	dbXML
Malo (1)	XpSQL	XpSQL	XpSQL	
	Oracle Clob	Oracle Clob	Oracle Clob	
	Oracle Clob	Oracle Clob	Oracle Clob	

La actualización muestra pobres rendimientos tanto para XpSQL como para dbXML. Oracle Clob presenta en general tiempos de respuesta buenos, pero no son comparados con los excelentes rendimientos que muestran Tamino y Oracle Mapeo OR tanto cuando aumenta la cantidad de archivos como cuando aumenta la complejidad de la estructura así como para la complejidad de la actualización. Los rendimientos para los tres tipos de actualización que se muestran en la Tabla 39 son los mismos.

Tabla 39. Rendimientos Update1, Update2 y Update3

		Nivel	Estructura		
			Simple (10 Elementos)	Mediana (50 - 100 Elementos)	Compleja (500 Elementos)
Cantidad de archivos	Baja (10 - 100 Archivos)	Mejor (5)	Oracle Mapeo OR Tamino	Oracle Mapeo OR Tamino	Oracle Mapeo OR Tamino
		Bueno (3)	dbXML	dbXML	dbXML
		Malo (1)	XpSQL Oracle Clob	XpSQL Oracle Clob	XpSQL Oracle Clob
	Media (100 - 1000 Archivos)	Mejor (5)	Oracle Mapeo OR Tamino	Oracle Mapeo OR Tamino	Oracle Mapeo OR Tamino
		Bueno (3)	dbXML	dbXML	dbXML
		Malo (1)	XpSQL Oracle Clob	XpSQL Oracle Clob	XpSQL Oracle Clob
	Alta (1000 - 5000 Archivos)	Mejor (5)	Oracle Mapeo OR Tamino	Oracle Mapeo OR Tamino	Oracle Mapeo OR Tamino
		Bueno (3)	dbXML	dbXML	dbXML
		Malo (1)	XpSQL Oracle Clob	XpSQL Oracle Clob	XpSQL Oracle Clob

Aún cuando los resultados obtenidos tienen rangos de validez acotados por los datos de ejemplo y la plataforma computacional usada, sí es posible extrapolarlos ya que los coeficientes de determinación (R^2) tienden a 1 en las pruebas, para resultados más concluyentes debieran hacerse más pruebas. También los rangos definidos para la complejidad de la estructura y la cantidad de archivos debiesen ser revisados.

Si bien Tamino, Oracle XMLDB y dbXML presentan rendimientos que se acomodan a las distintas operaciones, los rendimientos de XpSQL son bajos para todo tipo de operación en las distintas pruebas aplicadas, por lo tanto la única forma de que una organización escoja a XpSQL es cuando ésta tiene una base de datos PostgreSQL sobre Linux a la que se quiere agregar soporte para XML, en cualquier otro caso se debe descartar.

8. METODOLOGÍA

Después de analizar el funcionamiento de las distintas bases de datos en estudio, identificar los parámetros que más influyen en la elección de una base de datos y realizar un determinado número de pruebas y el análisis sobre los datos obtenidos, se pretende entregar una metodología que permita elegir una base de datos que mejor se acomode a las necesidades de una organización que trabaja con datos.

8.1. Formulación de la metodología.

Los siguientes pasos a seguir describen la metodología que se sugiere para la elección de la base de datos, esta metodología solamente evalúa el aspecto técnico ya que la decisión final depende de cuánto peso le asigne la organización a aspectos económicos o de otra índole:

1. Se debe solicitar a la organización la descripción de sus datos de acuerdo a la cantidad de archivos y a la complejidad de la estructura de los documentos XML, esta puede ser:

- Cantidad de Archivos: Baja - Media - Alta
 - Complejidad Estructura: Simple-Media- Compleja
2. Si la empresa u organización necesita conservar la estructura del documento XML sin modificaciones, por ejemplo por condiciones legales, se debe elegir entre:
 - Tamino – Oracle Clob – dbXML
 3. Si el costo es una condición importante de decisión se debe considerar:
 - Comerciales: Tamino – Oracle 10g (Clob, Mapeo OR)
 - Open Source: dbXML – XpSQL
 4. La empresa u organización debe distribuir 100 puntos para el almacenamiento, consulta y actualización de acuerdo a los tipos de operaciones que tienen más prioridad o frecuencia, para ello debe llenar la Tabla 40

Tabla 40. Distribución de puntos según tipo de operación

Tipo de Operación	Puntaje
Almacenamiento (ST)	
Consulta Simple (QS)	
Consulta Media- Compleja (QC)	
Actualización (UP)	
TOTAL	100

Se consideran las consultas de complejidad media y alta en la misma categoría debido a que los rendimientos de las bases de datos fueron los mismos en ambos casos. En la actualización sucede algo parecido, ya que para los tres niveles de complejidad las bases de datos respondieron de manera similar

5. Se debe elegir la columna de una de las siguientes tablas (Tabla 41, Tabla 42 y Tabla 43) que representan el peso asignado según el rendimiento de cada tipo de operación, de acuerdo a la cantidad de archivos y la complejidad de la estructura definida en el punto 1.

Tabla 41 Pesos asignados según el rendimiento en cada tipo de Operación para una cantidad de archivos baja.

Cantidad de archivos Baja	Tipo Operación	Estructura											
		Simple				Mediana				Compleja			
		ST	QS	QC	UP	ST	QS	QC	UP	ST	QS	QC	UP
Producto													
Tamino	3	5	5	5	3	5	5	5	3	5	5	5	
Oracle Mapeo OR	5	5	5	5	5	3	5	5	5	3	5	5	
Oracle Clob	5	1	1	3	5	1	1	3	5	1	1	3	
dbXML	5	3	3	1	5	5	3	1	5	5	3	1	
XpSQL	1	1	1	1	1	3	1	1	1	1	3	1	

Tabla 42 Pesos asignados según el rendimiento en cada tipo de Operación para una cantidad de archivos media.

Cantidad de archivos Media	Tipo Operación	Estructura											
		Simple				Mediana				Compleja			
		ST	QS	QC	UP	ST	QS	QC	UP	ST	QS	QC	UP
Producto													
Tamino	3	5	5	5	3	5	5	5	3	5	5	5	
Oracle Mapeo OR	5	5	5	5	5	3	5	5	5	5	5	5	
Oracle Clob	5	1	1	3	5	1	1	3	3	1	1	3	
dbXML	3	3	3	1	3	3	3	1	3	3	3	1	
XpSQL	1	1	3	1	1	1	3	1	1	3	3	1	

Tabla 43 Pesos asignados según el rendimiento en cada tipo de Operación para una cantidad de archivos alta.

Cantidad de archivos Alta	Tipo Operación	Estructura											
		Simple				Mediana				Compleja			
		ST	QS	QC	UP	ST	QS	QC	UP	ST	QS	QC	UP
Producto													
Tamino	5	5	5	5	5	5	5	5	5	5	5	5	
Oracle Mapeo OR	3	5	5	5	3	5	5	5	5	5	5	5	
Oracle Clob	5	1	1	3	5	1	1	3	3	1	1	3	
dbXML	3	3	3	1	3	3	3	1	3	3	3	1	
XpSQL	1	1	1	1	1	1	3	1	1	3	1	1	

Donde:

ST = Almacenamiento

QS = Consulta Simple

QC = Consulta Media- Compleja

UP = Actualización

- Se pondera el peso de cada rendimiento con los puntos distribuidos para cada tipo de operación en todas las bases de datos. La que obtenga el puntaje más alto será la base de datos más adecuada a elegir desde un punto de vista técnico:

$$\text{Base de Datos } i = RE * ST + RE * QS + RE * QC + RE * UP$$

9. CONCLUSIONES

Los sistemas de almacenamiento de datos proveen una solución a las crecientes demandas del manejo de datos XML, ya sea agregando nuevas capacidades a los productos existentes o creando nuevos productos especialmente diseñados para trabajar con XML. Estas diferentes tecnologías no se pueden considerar excluyentes, más bien son complementarias y permiten dar una solución integral al manejo de los datos XML.

Con el estudio de cada base de datos se pudo comprender cómo funcionan las operaciones básicas de almacenamiento, consulta y actualización para datos XML y además se aprendieron nuevas herramientas de administración (por ejemplo XPath, Xquery, Xupdate, etc.).

La metodología necesita que la organización determine la complejidad de los documentos XML que manejará y la cantidad de los mismos según unos rangos definidos en esta metodología, además de la frecuencia y prioridad de las operaciones básicas. Estos dos puntos son muy importantes ya que van a determinar en gran medida la elección de la base de datos que mejor responda a los requerimientos de esta organización.

De acuerdo al tipo de almacenamiento, podemos concluir que más que cómo se deben almacenar los datos, debemos preguntarnos qué tipo de operación es la que más se utiliza o qué aplicaciones van a procesar estos datos, no podemos afirmar que por el sólo hecho de que las aplicaciones procesen datos XML es preferible trabajar con bases de datos nativas debido a que se ahorra tiempo en las conversiones y se mantiene la fidelidad del documento, porque también algunos productos no nativos que presentan una gran robustez pueden minimizar estos tiempos a pesar de poseer una capa extra de transformaciones en el manejo de los documentos XML.

Finalmente como un trabajo a futuro se podría dar mayor validez estadística a los análisis probando en diferentes plataformas de hardware, eliminando las restricciones de capacidad, analizando una mayor cantidad de productos que presenten soporte para XML e incorporando aspectos económicos a la metodología.

REFERENCIAS

- [1] Lewis, W "Pillar of the Community," *Intelligence Enterprise* [Online], Feb. 2001. Disponible: <http://www.intelligenceenterprise.com/000818/feat1.html>
- [2] Bourret, R. "XML and Databases," [Online], Nov. 2003. Disponible: <http://www.rpbouret.com/xml/XMLAndDatabases.htm>
- [3] Dejesus, E. "XML Enters the DBMS Arena," *Computer World* [Online], Oct. 2000 Disponible: http://www.computerworld.com/cwi/story/0,1199,NAV63_STO53026,00.htm
- [4] Bray T., Paoli J. "Extensible Markup Language," *World Wide Web Consortium* [Online], Feb. 2004. Disponible: <http://www.w3.org/TR/2006/PER-xml-20060614/>