Construcción de sistemas de recuperación de información sobre córpora textuales estructurados de grandes dimensiones *

Fco. Mario Barcala
Centro Ramón Piñeiro
Santiago-Noia km. 3, A Barcia
36900 Santiago de Compostela
barcala@freeresearch.org

Miguel A. Molinero
Depto. de Informática
Universidade de Vigo
Campus As Lagoas, s/n
32004 Ourense
molinero@uvigo.es

Eva. Domínguez
Centro Ramón Piñeiro
Santiago-Noia km. 3, A Barcia
36900 Santiago de Compostela
edomin@cirp.es

Resumen: En este trabajo se evalúan las principales tecnologías para el desarrollo de sistemas de recuperación de información basados en córpora estructurados de grandes dimensiones: Oracle (Oracle Corporation, 8/3/2005) y Tamino (Software AG Company, 8/3/2005).
Palabras clave: Recuperación de información, córpora estructurados de grandes dimensiones, XML, Oracle, Tamino

Abstract: In this paper we evaluate main technologies to develop Information Retrieval Systems based on large text structured corpora: Oracle (Oracle Corporation, 8/3/2005) and Tamino (Software AG Company, 8/3/2005).
Keywords: Information Retrieval, large structured corpora, XML, Oracle, Tamino

1. Introducción

Normalmente los sistemas de recuperación de información convencionales trabajan con documentos que tienen una estructura muy elemental. Este tipo de sistemas consiguen resolver consultas de una manera muy eficiente, pero consideran siempre la totalidad de los documentos, y es imposible hacer la distinción entre diferentes secciones dentro de los mismos.

Por otra parte, los córpora textuales desarrollados actualmente se componen de un conjunto de documentos de texto que, de una u otra manera, tienen cierta complejidad estructural, por lo que la construcción de sistemas de recuperación de información convencionales no se beneficiará de ésta.

Existen diferentes ejemplos de sistemas de recuperación de información basados en córpora (López, 2003) (Davies, 2002), de métodos de búsqueda sobre córpora voluminosos (Davies, 2003), e incluso en (Akmal B. Chaudhri, 2003) se presenta una revisión de diferentes tecnologías que podrían ser utilizadas para construir sistemas de recuperación de información genéricos basados en la tecnología XML (World Wide Web Consortium, 8/3/2005). Sin embrago, hasta la fecha no se han realizado análisis o estudios comparativos sobre tecnologías y arquitecturas que puedan ser utilizadas para construir sistemas de recuperación de información que trabajen con córpora estructurados de grandes dimensiones.

Dado que los sistemas de recuperación de información pueden ser de muy diversa índole, en este caso nos centraremos en córpora que no incluyan ningún tipo de anotación morfosintáctica asociada a las palabras y que, además, estén estructurados en formato XML (World Wide Web Consortium, 8/3/2005). La mayoría de los aspectos aquí tratados serán válidos también para córpora anotados (aunque en estos casos hubiera que completar el estudio) o que no se encuentren en formato XML (ya que si los córpora están bien estructurados podrían convertirse a formato XML de una manera automática).

Por lo tanto, en primer lugar presentamos el corpus sobre el que se ilustrará el estudio. A continuación planteamos las principales alternativas para la construcción de sistemas de recuperación de información basados en córpora estructurados de grandes dimensiones. Posteriormente evaluamos las dos tecnologías que abanderan las líneas de investigación principales en este campo, definiendo

* Parcialmente financiado por el Ministerio de Educación y Ciencia (MEC) y FEDER (TIN2004-07246-C02-01 y TIN2004-07246-C02-02), por MEC (HF2002-81), y por la Xunta de Galicia (PGIDIT02PXIB30501PR, PGIDIT02SIN01E y PGIDIT03DIN30501PR)

previamente las necesidades para este tipo de sistemas: por un lado Oracle[1](Oracle Corporation, 8/3/2005), un gestor de bases de datos relacional que integra capacidades XML (*Extensible Markup Language*) (World Wide Web Consortium, 8/3/2005); por el otro, Tamino[2](Software AG Company, 8/3/2005), un indexador XML nativo. Finalmente, exponemos las conclusiones y la tecnología que mejor se adapta a las necesidades planteadas.

2. Definición del corpus objeto de estudio

A lo largo del presente trabajo se propondrán tecnologías, técnicas y métodos para construir sistemas de recuperación de información basados en córpora, los cuales serán ilustrados sobre un caso real que ha servido de ejemplo: es el "Corpus de Referencia do Galego Actual" (http://corpus.cirp.es/corgaxml). Éste, que en su versión XML consta actualmente de más de siete millones de formas repartidas en cientos de documentos, será también el utilizado para hacer la evaluación de las tecnologías empleadas.

Estos documentos pueden ser periódicos, revistas o libros, por lo que se dispone de una DTD (World Wide Web Consortium, 8/3/2005) para cada tipo de documento. A modo de ejemplo, en las figuras 1 y 2 se muestran las DTDs de periódicos y libros respectivamente [3].

3. Alternativas para la construcción

A la hora de construir un sistema de recuperación de información basado en córpora es importante tener en cuenta los siguientes aspectos:

- Es necesario separar el corpus en sí mismo del sistema de recuperación de información. Es decir, el sistema de recuperación de información no debe condicionar la estructura del corpus ni a la inversa, ya que se penalizará, bien la eficiencia del sistema de recuperación, bien la claridad de las DTDs del corpus.
- Utilizar diferentes estructuras en el sistema de recuperación de información para acomodar los diferentes tipos de documentos del corpus (periódicos, libros, etc.) redunda en una pérdida de rendimiento considerable a la hora de hacer consultas en el sistema. Cuando manejamos córpora de grandes dimensiones es necesario agrupar los diferentes tipos de documentos en una estructura común dentro del sistema de recuperación para optimizar su rendimiento.
- En este tipo de sistema prima la velocidad en la recuperación y no importa excesivamente la penalización en las inserciones o actualizaciones ya que, normalmente, estos sistemas se actualizan cada varios meses y no es muy importante que en esas fases de actualización el sistema invierta varias horas (o incluso días).

Actualmente, para la construcción de sistemas de recuperación de información sobre documentos en formato XML se están siguiendo dos líneas de investigación y desarrollo principales. Una consiste en adaptar los documentos XML al modelo relacional para poder introducir la información de éstos en un gestor de bases de datos relacional, y la otra se basa en introducir los documentos en un gestor XML nativo o que permita trabajar de alguna forma con documentos XML directamente (*XML-enabled*).

3.1. Gestor de base de datos relacional

Existen diferentes maneras de representar un documento XML de manera relacional (Akmal B. Chaudhri, 2003). Algunas de ellas son automáticas y otras necesitan un estudio previo para definir el modelo relacional que mejor se adapte al formato XML del corpus.

Para el caso de córpora de grandes dimensiones en los que se desee un alto rendimiento, las maneras de estructuración manuales *ad-hoc* permiten obtener un rendimiento mucho más elevado.

En la figura 3 se muestra un posible modelo entidad-relación que permite introducir los diferentes tipos de documentos del corpus de estudio en esa estructura común y que se ha utilizado para hacer la evaluación de esta tecnología.

[1]Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

[2]Tamino is a Software AG Product.

[3]Las DTDs utilizadas en realidad en el proyecto CORGA son algo más complejas. En este caso solo mostramos una adaptación simplificada de las mismas por motivos de claridad explicativa.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ENTITY % valores_distinto "otra_lengua|otro_período|no_normativo|desconocido">
<!ENTITY % texto "#PCDATA|referencia_nota">
<!ELEMENT documento (cabecera_documento, contenido_documento)>
<!ELEMENT cabecera_documento (identificador, medio, nombre,
          año_publicación, lugar_publicación, editorial)>
<!ELEMENT contenido_documento (sección+)>
<!ELEMENT sección (nombre, noticia+)>
<!ELEMENT noticia (cabecera_noticia, contenido_noticia)>
<!ELEMENT cabecera_noticia (identificador, autor+, área_temática+)>
<!ELEMENT contenido_noticia (titular?, resumen?, pié_de_foto*, cuerpo)>
<!ELEMENT titular (párrafo+)>
<!ELEMENT resumo (párrafo+)>
<!ELEMENT pié_de_foto (párrafo+)>
<!ELEMENT cuerpo (párrafo+, nota*)>
<!ELEMENT párrafo (oración+)>
<!ATTLIST párrafo distinto (%valores_distinto;) #IMPLIED>
<!ELEMENT oración (%texto;|distinto)*>
<!ATTLIST oración distinto (%valores_distinto;) #IMPLIED>
<!ELEMENT nota (párrafo+)>
<!ATTLIST nota identificador ID #IMPLIED
          tipo (referencia_bibliográfica) #IMPLIED>
<!ELEMENT referencia_nota EMPTY>
<!ATTLIST referencia_nota referencia IDREF #REQUIRED>
```

Figura 1: DTD de periódico. Los elementos que no están definidos son de tipo #PCDATA.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ELEMENT documento (cabecera_documento, contenido_documento)>
<!ELEMENT cabecera_documento (identificador, medio, título, autor+,
          año_publicación, editorial, área_temática+)>
<!ELEMENT contenido_documento (prólogo*, dedicatoria*, cita*, pié_de_foto*,
          cuerpo, apéndice*)>
<!ELEMENT prólogo (cabecera?, contenido)>
<!ELEMENT cabecera (autor+, área_temática*)>
<!ELEMENT contenido (encabezado?, pié_de_foto*, cuerpo)>
<!ELEMENT cuerpo (división+, nota*)>
<!ELEMENT apéndice (cabecera?, contenido)>
<!ELEMENT división (encabezado?, dedicatoria*, cita*, pié_de_foto*,
          cuerpo_división)>
<!ATTLIST división tipo (capítulo|parte) #IMPLIED>
<!ELEMENT encabezado (párrafo+)>
<!ELEMENT cuerpo_división (párrafo+ | división+)>
<!ELEMENT dedicatoria (párrafo+)>
<!ELEMENT cita (párrafo+)>
```

Figura 2: DTD de libro. Los elementos que no están definidos se corresponden con los de la DTD de periódico.

3.2. Gestor XML nativo

En el caso de utilizar un gestor XML nativo también es necesario definir un XML común que dé cabida a los diferentes tipos de documentos, y así evitar tener que consultar diferentes colecciones de documentos con la consiguiente pérdida de rendimiento.

Este formato común puede definirse de muchas maneras, pero una buena directiva a seguir consiste en que cuanto más homogéneo y más sencillo sea, mayor será el rendimiento que se obtendrá, ya que las consultas también involucrarán menos elementos estructurales. En la figura 4 se muestra una posible DTD que define este formato XML común que se ha utilizado para realizar la evaluación.

4. Evaluación

La evaluación se realizó tomando dos claros exponentes en cada una de las tecnologías objeto del estudio: por un lado Oracle (versión 9ir2), que es un gestor de base de datos relacional muy extendido en la actualidad, y por el otro Tamino (versión 4.2.1), que es un gestor XML nativo que presenta multitud de

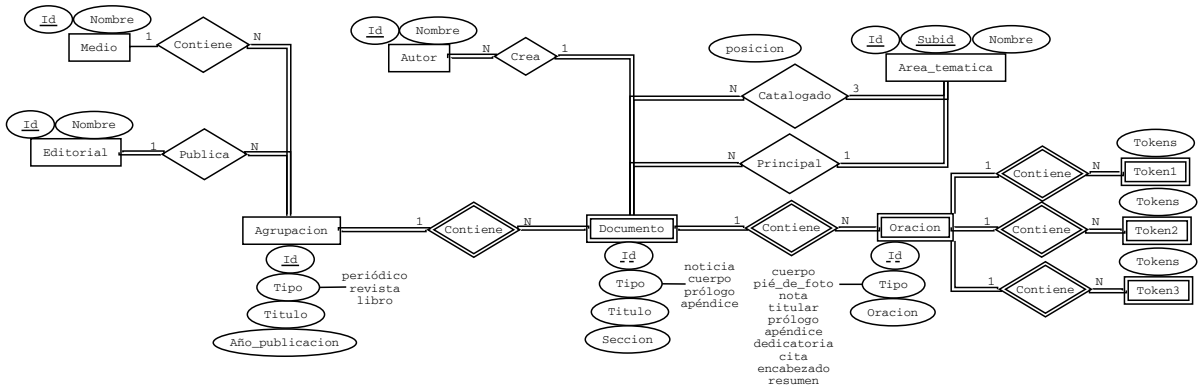

Figura 3: Posible modelo entidad-relación.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ELEMENT agrupación (cabecera_agrupación, documento+)>
<!ELEMENT cabecera_agrupación (título, medio, año_publicación,
          editorial, autor*, área_temática*)>
<!ELEMENT documento (autor+, área_temática+, oración+)>
<!ATTLIST documento tipo (Noticia|Prólogo|Apéndice|Cuerpo) #REQUIRED
             sección CDATA #IMPLIED
                     título CDATA #IMPLIED>
<!ENTITY % texto "#PCDATA|referencia_nota">
<!ELEMENT oración (texto_oración, tokens)>
<!ATTLIST oración
   tipo (Prólogo|Apéndice|Titular|Resumen|Pié_de_foto|Cuerpo|Nota|
                Encabezado|Dedicatoria|Cita) #REQUIRED>
<!ELEMENT texto_oración (%texto;|distinto)*>
<!ELEMENT tokens (token1+, token2*, token3*)>
<!ELEMENT token1 (#PCDATA)>
<!-- Igual para token2 y token3-->
<!ELEMENT distinto (%texto;)*>
```

Figura 4: Posible DTD común. Los elementos que no están definidos son de tipo #PCDATA.

capacidades.

Para la evaluación se utilizaron dos criterios clave:

- **Flexibilidad**: A partir de los diferentes requisitos identificados que necesitan los sistemas de recuperación de información basados en córpora, se determina en qué medida cada una de las tecnologías probadas verifica cada uno de éstos.
- **Rendimiento**: Utilizando las consultas más representativas en este tipo de sistemas (que sirven de modelo), se evalúa el rendimiento que se obtiene en la ejecución de cada una de ellas.

  Aunque existen diferentes bancos de pruebas (*benchmarks*) sobre sistemas de recuperación XML (Akmal B. Chaudhri, 2003), éstos son demasiado genéricos y no están orientados a las necesidades de los sistemas basados en córpora, por lo que se ha optado por diseñar las consultas tipo que se utilizarán en este tipo de sistemas. Además, los bancos de pruebas no suelen servir para diferentes tipos de tecnologías, es decir, existen bancos de pruebas para gestores relacionales y otros diferentes para indexadores XML nativos, pero no bancos que determinen el rendimiento en las dos tecnologías a la vez.

4.1. Flexibilidad

Los sistemas de recuperación de información basados en córpora pueden tener objetivos bien diferentes, y el tipo de consultas que se realicen varía de un sistema a otro, pero en general, presentan las siguientes necesidades genéricas:

1. **Estadísticas**: Obtener valores numéricos a diferentes niveles. Por ejemplo, contar el número de casos y de documentos que satisfacen una consulta.

2. **Información adicional**: Ofrecer información adicional en los resultados. Por ejemplo, devolver las secuencias de términos que satisfacen un criterio de búsqueda, pero adicionalmente, también mostrar el autor del documento relevante, la editorial, etc. es decir, datos adicionales que se encuentran en la estructura de los documentos del corpus.

3. **Resaltado**: Posibilitar la identificación del término o términos que causaron la coincidencia y resaltarlos dentro de los resultados obtenidos.

4. **Contexto**: Mostrar un contexto variable con respecto a las cadenas de búsqueda. Por ejemplo, en el caso de estudio debe ser posible mostrar toda la oración donde se encuentra algún caso de búsqueda, o n oraciones antes y n después, o incluso n palabras antes y n después dentro de la misma oración con respecto al término que ha presentado coincidencia.

5. **Tipos de indexación/métodos de busca**: Entre los aspectos a tener en cuenta en cuanto a las posibilidades de indexación y búsqueda destacan:

   a) Permitir búsquedas por coincidencia exacta.

   b) Permitir búsquedas sensibles o no a los acentos.

   c) Permitir búsquedas sensibles o no a mayúsculas.

   d) Permitir búsquedas booleanas.

   e) Permitir la utilización de operadores de proximidad.

   f) Permitir varios criterios de *tokenización* e indexación en la misma base documental. Por ejemplo, construir dos índices de texto, uno sensible a los acentos y el otro no, y que el usuario pueda decidir cual es el que utiliza a la hora de realizar la consulta.

   g) Excluir cierto texto marcado del índice o incluso indexar aparte algunas porciones correctamente delimitadas. Por ejemplo, posibilitar la exclusión del índice de palabras en otro idioma que aparecen englobadas dentro de alguna etiqueta, o poder buscar nombres propios delimitados en el texto.

   h) Ignorar ciertos caracteres en la indexación. En ocasiones los documentos del corpus tienen ciertos caracteres utilizados para marcar diferentes aspectos. En estos casos muchas veces resulta necesario poder indexar las palabras sin estos caracteres pero visualizarlas con ellos, es decir, ignorar esos caracteres a la hora de construir el índice.

6. **Codificación de caracteres**: Soportar diferentes tipos de codificación de caracteres (ASCII, ISO-8859-1, UTF-8, etc.).

7. **Utilización de comodines**: Permitir sustituciones de caracteres, expresiones regulares, etc.

8. **Mostrado de resultados y navegación**: Navegar a través de los resultados, mostrándolos página a página (sin tener que rehacer la consulta).

9. **Ordenación**: Utilizar varios criterios de ordenación simultáneamente.

10. **Relaciones estructurales**: Se refiere a la flexibilidad en las consultas cuando se incluyen restricciones estructurales, es decir, cuestiones como las capacidades del lenguaje de consulta y limitaciones en la complejidad del mismo.

Evaluación

En la tabla de la figura 4.1 se muestra si las tecnologías soportan cada uno de los elementos definidos en el punto anterior. Para el caso de Tamino se incluyen dos columnas, una para cada uno de los lenguajes que soporta: XQuery, una propuesta de estándar del *World Wide Web Consortium* (World Wide Web Consortium, 8/3/2005) y X-Query un lenguaje propietario de Tamino.

A la vista de estos datos, las apreciaciones que podemos extraer son las siguientes:

- El lenguaje X-Query es muy poco flexible comparado con las otras dos alternativas, por lo que en las apreciaciones subsiguientes sólo se tendrá en cuenta el lenguaje XQuery cuando nos refiramos a Tamino.
- Tanto Oracle como Tamino permiten obtener estadísticas a cualquier nivel siempre que ese nivel esté estructurado. Esto

	Oracle	Tamino XQuery	Tamino X-Query
1	varias al mismo tiempo (tokens necesarios)	una de cada vez (tokens necesarios)	**No**
2	Sí	Sí	limitada
3	Sí	Sí	**No**
4	Sí (varias posibilidades)	Sí	**No**
5a	Sí	Sí	Sí
5b	Sí	Sí	Sí
5c	Sí	**No**	**No**
5d	Sí	Sí	**No**
5e	NEAR	PROXIMITY-CONTAINS	ADJ,NEAR
5f	Sí	**No**	**No**
5g	Sí	Sí	Sí
5h	SKIPJOINS	**No**	**No**
6	Sí	Sí	Sí
7	%,_	*,?	*,?
8	Sí	Sí	Sí
9	Sí (varios criterios)	Sí (varios criterios)	(varios criterios)
10	SQL (flexibilidad alta)	XQuery (muy alta)	X-Query (baja)

Figura 5: Evaluación de la flexibilidad.

puede ser problemático porque puede ser necesario repetir información o estructurar información excesivamente.

Por ejemplo, en el modelo entidad-relación de la figura 3 podemos comprobar que un documento se va descomponiendo hasta el nivel de palabra (*token*). Si consideramos que *token1*, *token2* y *token3* contendrán las secuencias de una, dos y tres palabras de la oración respectivamente, puede comprobarse que el texto de las oraciones se encuentra repetido tanto en la tabla "oración" como en las tablas "*token*". Esta estructura es necesaria para realizar todos los tipos de estadísticas planteados anteriormente (en concreto, para poder contar el número de apariciones de expresiones de hasta tres palabras).

Por otra parte, solo Oracle permite obtener varias estadísticas de una vez compartiendo los cálculos computacionales, mejorando considerablemente el rendimiento de ese tipo de consultas.

- Oracle ofrece más posibilidades a la hora de mostrar el contexto de los términos de búsqueda.
- Ambas tecnologías incluyen operadores de proximidad similares, pero Tamino no permite crear índices sensibles a mayúsculas, ni definir diferentes criterios de *tokenización* e indexación para la misma base de datos, ni tampoco ignorar ciertos caracteres en la indexación.
- Tanto Tamino como Oracle soportan diversos tipos de codificación de caracteres, incluyendo UTF-8.
- Ambas tecnologías permiten varios criterios de ordenación simultáneos.
- XQuery es incluso más flexible que SQL, ya que presenta una sintaxis mucho más compleja que permite manipular y representar estructuras más elaboradas.

  Aunque se podría pensar que son pocas las necesidades que no cumple Tamino, debemos tener en cuenta que las necesidades de los sistemas de recuperación de información basados en córpora mostradas anteriormente, son necesidades mínimas en muchos de estos sistemas, y el incumplimiento de alguno de ellos puede suponer la diferencia entre un sistema útil y otro inútil.

  Si a esto añadimos que XQuery es aún una propuesta de estándar y que en Tamino aún no está completamente implementado, Oracle parece una alternativa

clara para la construcción de este tipo de sistemas desde el punto de vista de la flexibilidad.

4.2. Rendimiento

Para evaluar el rendimiento, en primer lugar definiremos varias consultas que cubren diferentes aspectos relacionados con los puntos anteriores, y a continuación mostraremos los tiempos obtenidos con cada una de las tecnologías elegidas.

Estas consultas se basan en el corpus objeto de estudio, pero los aspectos tratados son suficientemente genéricos como para ser aplicables a cualquier sistema de recuperación sobre córpora textuales estructurados.

Definición de las consultas

A continuación se definen cada una de las consultas junto con el tipo de medición que se realiza para ellas:

Q1 Obtención del número de documentos y de casos de las oraciones que contienen la expresión "sen embargo" para cada uno de los medios, de las áreas temáticas y de los lustros contemplados en el sistema.

El sistema considera tres tipos de medios, seis áreas temáticas y seis lustros, por lo que deben obtenerse treinta valores numéricos que deben mostrarse conjuntamente. Se mide el tiempo que transcurre desde que se hacen las consultas hasta que se recorren todos los resultados para su visualización.

Q2 Obtención de las oraciones correspondientes a la consulta Q1, obteniendo para cada una de las oraciones, el título del documento, la editorial, los autores, el medio, las áreas temáticas, el año de publicación y el tipo de oración que es. Las palabras buscadas aparecerán resaltadas.

Se mide el tiempo desde que se inicia la consulta hasta que se muestran todas las oraciones junto con los valores relacionados. Se trata de ver la eficiencia en la obtención de los valores asociados a las oraciones y el resaltado de los términos de búsqueda.

Q3a Obtención del número de documentos y de casos de las oraciones que contienen alguna palabra con el prefijo "pre".

Se trata de ver la eficiencia del índice de texto en cuanto a prefijos.

Q3b Obtención del número de documentos y de casos de las oraciones que contienen alguna palabra con el sufijo "ado".

Se trata de ver la eficiencia del índice de texto en cuanto a sufijos.

Q4 Obtención del número de documentos y de casos de las oraciones que contienen alguna palabra con el prefijo "in" o "im", pero solamente en los titulares de noticias de los periódicos.

Se combina una consulta computacionalmente costosa con varios filtros estructurales para obtener una estimación del coste de dichos filtros en el rendimiento de las consultas.

Evaluación

Las pruebas se realizaron en un PC Intel Celeron a 2.4GHz con 512 Mb DDR, un disco IDE de 40Gb y Windows XP Professional[4]. El subconjunto de documentos de prueba está formado por 600.000 palabras repartidas uniformemente entre periódicos, revistas y libros.

Aunque efectivamente quizás no sea el hardware más adecuado para la explotación de estos sistemas (donde se necesitan servidores potentes para manejar decenas o centenas de millones de palabras), sí que nos permite comprobar cómo hay una diferencia considerable de rendimiento entre las dos tecnologías contempladas, tal y como se puede apreciar en la figura 6.

Entre los resultados obtenidos destaca la diferencia de tiempos obtenida para la primera consulta, debido a que Tamino no puede calcular varios valores estadísticos compartiendo los cálculos para obtener un rendimiento óptimo.

Nuevamente quedan patentes las diferencias entre Oracle y Tamino, confirmando la tecnología Oracle como la primera opción para el desarrollo de estos sistemas.

5. Conclusiones

En primer lugar es necesario resaltar que, cuando se manejan volúmenes de información muy elevados, es imprescindible convertir los documentos del corpus a una estructura común, para obtener así un alto rendimiento. Mantener estructuras diferentes para cada

[4]Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Consulta	Oracle (9ir2)	Tamino XQuery(4.2.1)
Q1	10,891s	585,203s
Q2	5,922s	33,969s
Q3a	5,828s	106,719s
Q3b	34,172s	118,266s
Q4	12,547	93,172

Figura 6: Evaluación del rendimiento.

tipo de documento acarrea numerosas dificultades tanto de rendimiento como de gestión de los resultados.

Por otra parte, actualmente las tecnologías para la construcción de sistemas de recuperación de información no están preparadas para satisfacer las necesidades de los usuarios de sistemas basados en córpora. Por lo tanto, es necesario que, en el futuro, se desarrollen nuevos añadidos que tengan en cuenta estos sistemas.

Además, estas tecnologías están pensadas para sistemas que se puedan actualizar, y se necesitan más opciones de optimización para sistemas que prioricen drásticamente la eficiencia en las consultas frente a la de las actualizaciones, como es el caso que nos ocupa.

Aunque efectivamente las tecnologías XML se están desarrollando muy rápidamente, aún deben asentarse. Esta evolución demasiado acelerada hace complicada la robustez y el desarrollo claro de aplicaciones, dificultando así la puesta en marcha de estas tecnologías para sistemas en explotación.

Finalmente, los gestores relacionales presentan una alta robustez, flexibilidad y rendimiento fruto de la larga vida que mantienen en el mercado. Teniendo en cuenta los datos que arrojan las pruebas realizadas, que estos gestores incluyen en la actualidad capacidades de indexación de texto, y las limitaciones que presentan en general todas las tecnologías para la consideración de las necesidades de los sistemas basados en córpora, es lógico que en este momento propongamos Oracle como primera opción para construirlos.

Bibliografía

Akmal B. Chaudhri, Awais Rashid, Roberto Zicari. 2003. *XML Data Management, Native XML and XML-Enabled Database Systems.* Addison-Wesley.

Davies, Mark. 2002. Un corpus anotado de 100.000.000 palabras del español histórico y moderno. páginas 21–27, Valladolid, España.

Davies, Mark. 2003. Relational n-gram databases as a basis for unlimited annotation on large corpora. páginas 23–33, Lancaster, England.

López, María Sol. 2003. Corga (corpus de referencia del gallego actual). páginas 500–504.

Oracle Corporation. 8/3/2005. http://www.oracle.com.

Software AG Company. 8/3/2005. http://www.softwareag.com.

World Wide Web Consortium. 8/3/2005. http://www.w3c.org.