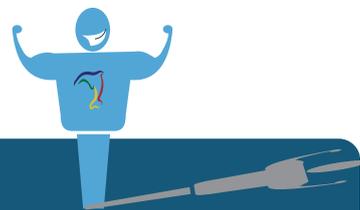


¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

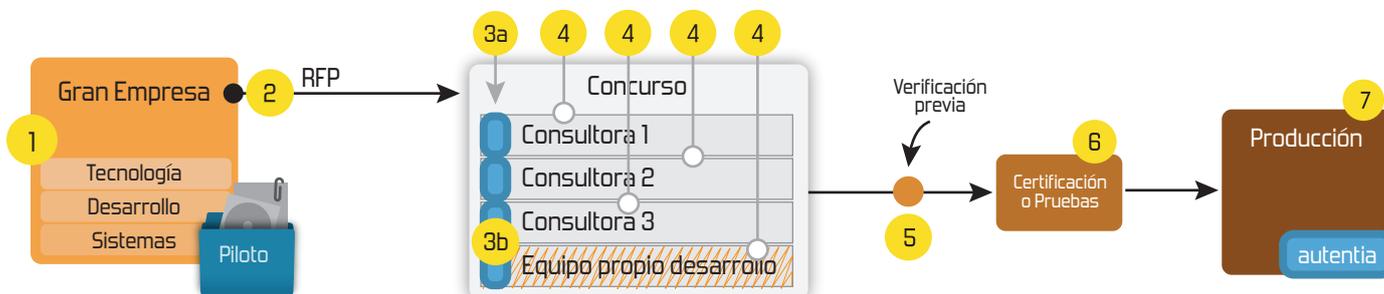
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)


 » Estás en: [Inicio](#) [Tutoriales](#) [Introducción a XQuery con ejemplos](#)

Rubén Aguilera Díaz-Heredero

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática, especialidad en Ingeniería del Software

 Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)

Catálogo de servicios Autentia



Fecha de publicación del tutorial: 2013-08-06

 Tutorial visitado 1 veces [Descargar en PDF](#)

Introducción a XQuery con ejemplos

0. Índice de contenidos.

- 1. Entorno
- 2. Introducción
- 3. Creación y configuración de la base de datos de ejemplo
- 4. Ejemplos
- 5. Conclusiones

1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Mac Book Pro 17" (2,6 Ghz Intel Core i7, 8 GB DDR3)
- Sistema Operativo: Mac OS X Snow Leopard 10.6.4
- XQuery 1.0

2. Introducción

XQuery es un lenguaje de consulta que proporciona los medios para extraer y manipular información de cualquier fuente de datos que pueda ser representada en formato XML.

Para su funcionamiento interno se apoya en XPath para poder acceder a la información contenida en nodos del documento XML. La otra utilidad fundamental de este lenguaje es que nos da la posibilidad de transformar un XML en otro distinto o incluso en otro formato. Un ejemplo muy típico es tener una serie de datos en XML y generar un XHTML para que puedan ser mostrados en un navegador.

En este tutorial vamos a empezar a gatear con el lenguaje y vamos a ver cómo recuperar información de un XML a través de varios ejemplos.

3. Creación y configuración de la base de datos de ejemplo

Vamos a trabajar con XML. Por lo tanto lo que primero necesitamos es un fichero .XML con el siguiente contenido:

```

view plain print ?
01. <tutoriales>
02.   <tutorial anio="2007">
03.     <titulo>XMLBeans, una forma de mapear un XML en objetos Java</titulo>
04.     <autor>
05.       <nombre>Alejandro</nombre>
06.       <apellidos>Pérez García</apellidos>
07.     </autor>
08.     <categoria>XML</categoria>
09.     <visitas>25827</visitas>
10.   </tutorial>
11.   <tutorial anio="2013">
12.     <titulo>Spring Security: haciendo uso de un servidor LDAP embebido</titulo>
13.     <autor>
14.       <nombre>José Manuel</nombre>
15.       <apellidos>Sánchez Suárez</apellidos>
16.     </autor>
17.     <categoria>Spring</categoria>
18.     <visitas>1996</visitas>
19.   </tutorial>
20.   <tutorial anio="2011">
21.     <titulo>Mybatis con Maven y Spring</titulo>
22.     <autor>
23.       <nombre>Rubén</nombre>
24.       <apellidos>Aguilera Díaz-Heredero</apellidos>
25.     </autor>
26.     <categoria>Mybatis</categoria>
  
```



Síguenos a través de:



Últimas Noticias

- » Técnicas de división de historias de usuario
- » Dolomitas on Giro
- » Comentando el libro: Agile Management de Angel Medinilla
- » Final de temporada de Terrakas Autentia freelances
- » Atención, APLAZADO Estreno último capítulo de Terrakas

[Histórico de noticias](#)

Últimos Tutoriales

- » Contando mi experiencia en los talleres gratuitos de Apple
- » Comentando el libro: Disciplined Agile Delivery (DaD)
- » Crear una rutina propia en Talend
- » Spring Container y la Inyección de Dependencias
- » Introducción a Spring Batch

```

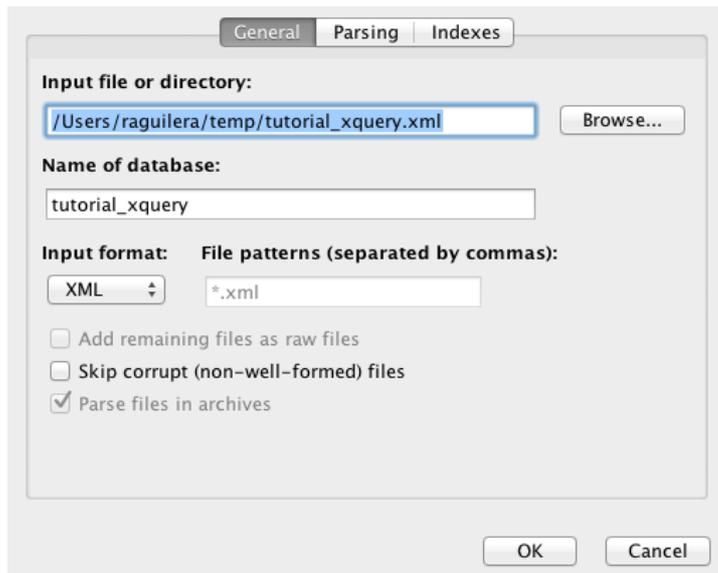
27.     <visitas>6998</visitas>
28. </tutorial>
29. <tutorial anio="2011">
30.     <titulo>Trabajar con XML sabiendo lo mínimo</titulo>
31.     <autor>
32.         <nombre>Roberto</nombre>
33.         <apellidos>Canales Mora</apellidos>
34.     </autor>
35.     <categoria>XML</categoria>
36.     <visitas>34842</visitas>
37. </tutorial>
38. <tutorial anio="2013">
39.     <titulo>Introducción a Spring Batch</titulo>
40.     <autor>
41.         <nombre>Miguel</nombre>
42.         <apellidos>Arlandy Rodríguez</apellidos>
43.     </autor>
44.     <categoria>Spring</categoria>
45.     <visitas>1273</visitas>
46. </tutorial>
47. </tutoriales>

```

Almacenamos el fichero con el nombre que queramos (ej. tutorial_xquery.xml)

Para implementar y probar los ejemplos contra esta base de datos vamos a hacer uso del editor BaseX que podéis descargar [aquí](#)

En mac, una vez descargado, descomprimos el .zip y ejecutamos el fichero BaseX.app. Para poder hacer uso del fichero XML creado anteriormente tenemos que ir a las opciones de menú "Database" --> "New" y seleccionar el fichero XML.



Pulsamos en "OK" y ya tenemos la base de datos en XML preparada para seguir los ejemplos.

4. Ejemplos

En todos los ejemplo vamos a hacer referencia al XML de prueba haciendo uso de la función de XQuery fn:doc(nombre_basedatos). En nuestro caso, fn:doc("tutorial_query").

Devolver todos los elementos que cuelgan de un nodo sin incluirlo

En esta expresión utilizamos el * para indicar que queremos todos los elementos que cuelgan de un nodo pero que no queremos incluir el nodo mismo. Este sería el ejemplo para devolver la información de todos los tutoriales.



Devolver todos los elementos que cuelgan de un nodo incluyéndolo

Últimos Tutoriales del Autor

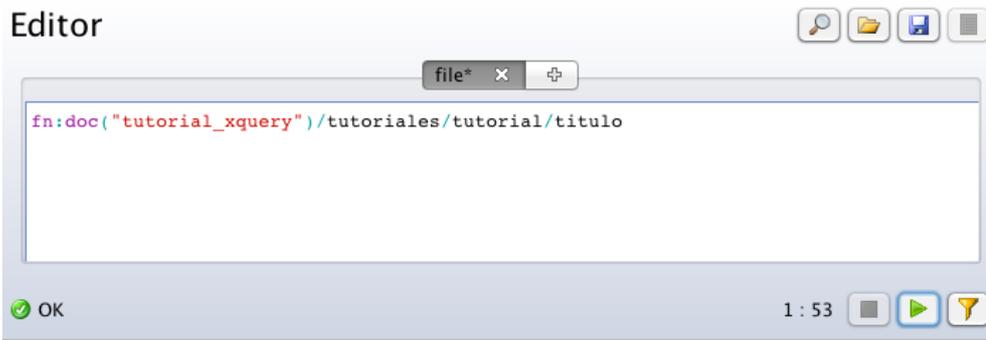
- » Crear una rutina propia en Talend
- » Clustering de puntos en Google Maps v3
- » Resolver problema LockTimeoutException en Spring Web Flow
- » Trabajar con tablas en JasperReport
- » Aspectos avanzados de Sitemesh

Últimas ofertas de empleo

- 2011-09-08
Comercial - Ventas - MADRID.
- 2011-09-03
Comercial - Ventas - VALENCIA.
- 2011-08-19
Comercial - Compras - ALICANTE.
- 2011-07-12
Otras Sin catalogar - MADRID.
- 2011-07-06
Otras Sin catalogar - LUGO.

Para esta expresión no hacemos uso del *. Tenemos que poner el path absoluto del nodo que queramos consultar. Este sería el ejemplo para devolver el título de todos los tutoriales.

Editor

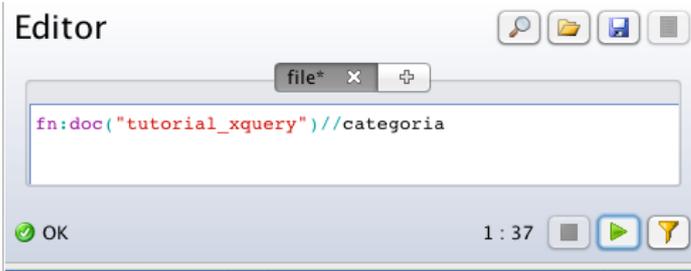


Text

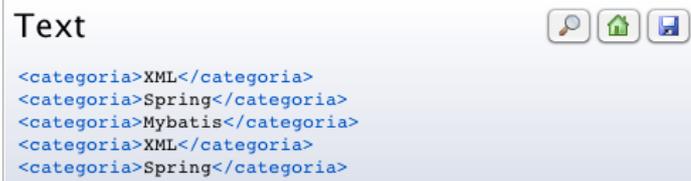


Una forma de acceder directamente a un nodo sin necesidad de escribir todo su path absoluto es con la notación `"/"` seguida del nodo.

Editor



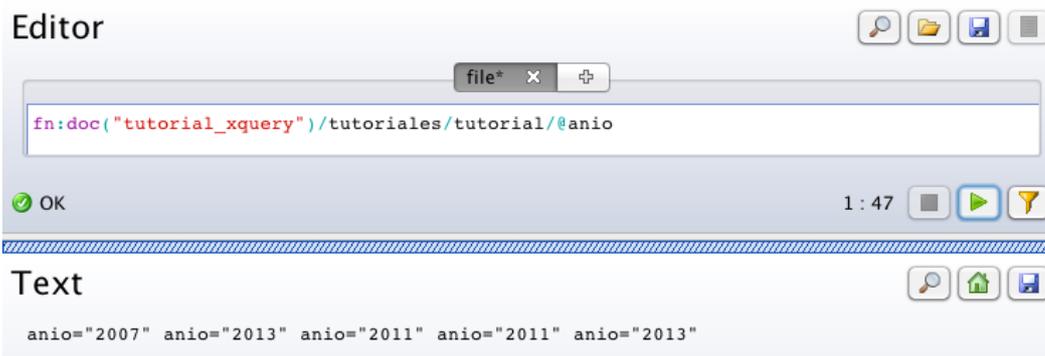
Text



Devolver atributos ligados a un nodo

Con esta expresión accedemos a los atributos que están ligados a un nodo. Este sería el ejemplo para acceder a la información del año de publicación de cada tutorial.

Editor



Text



Devolver elementos que cumplan una determinada condición

Podemos establecer una condición en el atributo de un nodo para que solo se devuelvan los que cumplan con esa condición. Este sería el ejemplo para devolver solo los tutoriales cuyo año de publicación sea menor que "2010".

Editor



file* x +

```
fn:doc("tutorial_xquery")/tutoriales/tutorial[@anio < 2010]
```

OK

1 : 59



Text



```
<tutorial anio="2007">
  <titulo>XMLBeans, una forma de mapear un XML en objetos Java</titulo>
  <autor>
    <nombre>Alejandro</nombre>
    <apellidos>Pérez García</apellidos>
  </autor>
  <categoria>XML</categoria>
  <visitas>25827</visitas>
</tutorial>
```

Igualmente podemos establecer una condición para que devuelva solo determinados nodos. Este sería el ejemplo para devolver solo los tutoriales que sean de la categoría "Mybatis".

Editor



file* x +

```
fn:doc("tutorial_xquery")/tutoriales/tutorial[categoria = "Mybatis"]
```

OK

1 : 67



Text



```
<tutorial anio="2011">
  <titulo>Mybatis con Maven y Spring</titulo>
  <autor>
    <nombre>Rubén</nombre>
    <apellidos>Aguilera Díaz-Heredero</apellidos>
  </autor>
  <categoria>Mybatis</categoria>
  <visitas>6998</visitas>
</tutorial>
```

También podemos hacer estas condiciones todo lo complejas que necesitemos gracias a las cláusulas and y or. Este sería el ejemplo que devuelve los tutoriales que sean de la categoría "XML" y estén escritos en el año "2007"

Editor



file* x +

```
fn:doc("tutorial_xquery")/tutoriales/tutorial[@anio="2007" and
categoria = "XML"]
```

OK

1 : 58



Text



```
<tutorial anio="2007">
  <titulo>XMLBeans, una forma de mapear un XML en objetos Java</titulo>
  <autor>
    <nombre>Alejandro</nombre>
    <apellidos>Pérez García</apellidos>
  </autor>
  <categoria>XML</categoria>
  <visitas>25827</visitas>
</tutorial>
```

Podemos utilizar otros operandos como: "=", "<", "<=", ">", ">="

Acceso directo a un nodo de una lista

Si conocemos el orden de los elementos de una XML puede ser útil poder acceder directamente a uno de ellos por posición. Este sería el ejemplo que nos devuelve la información del tutorial que se encuentra en la posición 5 de la lista de tutoriales.

Editor

file* x +

```
fn:doc("tutorial_xquery")/tutoriales/tutorial[5]
```

OK 1 : 49

Text

```
<tutorial anio="2013">
  <titulo>Introducción a Spring Batch</titulo>
  <autor>
    <nombre>Miguel</nombre>
    <apellidos>Arlandy Rodríguez</apellidos>
  </autor>
  <categoria>Spring</categoria>
  <visitas>1273</visitas>
</tutorial>
```

Búsquedas textuales

Podemos realizar consultas que incluyan búsquedas textuales sobre un elemento en concreto. Este sería el ejemplo que devuelve los tutoriales donde la palabra "Maven" está contenida dentro de su título.

Editor

file* x +

```
fn:doc("tutorial_xquery")/tutoriales/tutorial[contains(titulo, "Maven")]
```

OK 1 : 65

Text

```
<tutorial anio="2011">
  <titulo>Mybatis con Maven y Spring</titulo>
  <autor>
    <nombre>Rubén</nombre>
    <apellidos>Aguilera Díaz-Heredero</apellidos>
  </autor>
  <categoria>Mybatis</categoria>
  <visitas>6998</visitas>
</tutorial>
```

Devolver el nodo padre de un elemento

Para devolver el nodo padre de un elemento utilizamos la notación "..". Este sería el ejemplo para devolver la información del tutorial donde el nombre del autor sea "Roberto".

Editor

file* x +

```
fn:doc("tutorial_xquery")/tutoriales/tutorial/autor[nombre = "Roberto"]/..
```

OK 1 : 70

Text

```
<tutorial anio="2011">
  <titulo>Trabajar con XML sabiendo lo mínimo</titulo>
  <autor>
    <nombre>Roberto</nombre>
    <apellidos>Canales Mora</apellidos>
  </autor>
  <categoria>XML</categoria>
  <visitas>34842</visitas>
</tutorial>
```

5. Conclusiones

Hasta aquí hemos visto como podemos realizar todo tipo de consultas sobre datos estructurados en XML. En posteriores tutoriales veremos como transformar el XML en otro que se ajuste a nuestras necesidades y más utilidades de este lenguaje.

Cualquier duda o sugerencia en la zona de comentarios.

Saludos.

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

[Share](#) |

0

Animáte y coméntanos lo que pienses sobre este **TUTORIAL**:

» **Regístrate** y accede a esta y otras ventajas «



Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

IMPULSA

Impulsores

Comunidad

[¿Ayuda?](#)

0 personas han traído clicks a esta página

sin clicks

+ + + + + + + +

powered by [karmacacy](#)

Copyright 2003-2013 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

