



Linkia FP

Formación Profesional Oficial a Distancia

DAM – M06 – Clase 09

Acceso a datos

BBDD nativas en XML

CLASE

Contenido del módulo

- UF3: Bases de datos nativas en XML.
 - eXist.

Tema 5
3 Clases
1 Actividad

Plan de trabajo

Plan de trabajo - Curso 2021/2022

DAM - M06: Acceso a datos

UF	FECHAS	TEMAS	Recursos complementarios: Video clases	EVALUACIONES				EXAMEN	
				ACTIVIDADES DE EVALUACIÓN CONTÍNUA					
				Entregas o participaciones 70%	Clases 10%*	Test 20%			
UF1: Persistencia en ficheros	02/02/2022 - 14/02/2022	Tema 1. Persistencia en ficheros	VC01	Actividad 1	C01 C02	Test UF1	Examen UF1		
	14/02/2022 - 28/02/2022	Tema 2. Persistencia en BDR con JDBC	VC02		C03 C04	Test UF2	Examen UF2		
	28/02/2022 - 14/03/2022	Tema 3. Persistencia BDR con ORM	VC03	Actividad 2	C05 C06				
	14/03/2022 - 28/03/2022	Tema 4. Persistencia en BDOO – BDOR	VC04	Actividad 3	C07 C08				
UF3: Persistencia en BD Nativas XML	28/03/2022 - 19/04/2022	Tema 5. Persistencia en BBDD nativas XML	VC05	Actividad 4	C09 C10 C11	Test UF3	Examen UF3		
UF4: Componentes de acceso a datos	19/04/2022 - 02/05/2022	Tema 6. Componentes da acceso a datos	VC06	Actividad 5	C12 C13	Test UF4	Examen UF4		
	02/05/2022 - 16/05/2022				C14				

Horario de las clases

NÚMERO CLASE	DÍA DE LA SEMANA	FECHA	HORA INICIO
C01	MARTES	08/02/2022	15:15
C02	VIERNES	11/02/2022	14:30
C03	VIERNES	18/02/2022	14:30
C04	VIERNES	25/02/2022	14:30
C05	VIERNES	04/03/2022	14:30
C06	VIERNES	11/03/2022	14:30
C07	VIERNES	18/03/2022	14:30
C08	VIERNES	25/03/2022	14:30
C09	VIERNES	01/04/2022	14:30
C10	MARTES	05/04/2022	15:15
C11	VIERNES	08/04/2022	14:30
C12	VIERNES	22/04/2022	14:30
C13	VIERNES	29/04/2022	14:30
C14	VIERNES	06/05/2022	14:30

Contenido clase

UF3: Bases de datos nativas en XML.

- BBDD nativas XML.
- Estrategias de almacenamiento.
- Librerías de acceso a datos XML.
- eXist.
- Establecimiento y cierre de conexiones.
- Creación y borrado de recursos: clases y métodos.
- Modificación de contenidos XML.



BBDD XML vs BBDDR

- El modelo de datos XML es muy diferente a otros modelos subyacentes en sistemas gestores más tradicionales.
- En un modelo relacional, los datos se almacenan siguiendo una estructura.
- Sin embargo, en un modelo de datos XML, esta estructura no es tan estricta, sino que es un semi-estructura → Documento.

¿Alternativas para almacenar XML?

- La primera alternativa es almacenar los documentos XML en un sistema gestor relacional o de objetos. Es lo que conocemos como alternativa híbrida.
 - Esta es la alternativa que utilizan actualmente, por ejemplo, los sistemas gestores de contenido (CMS) tipo Joomla o Drupal.

¿Alternativas para almacenar XML?

- La segunda alternativa es utilizar sistemas gestores nativos XML.
 - Con los sistemas XML nativos podemos almacenar tanto documentos centrados en los datos como centrados en el contenido, sin necesidad de perder datos en mapeos, como sucede en la alternativa anterior.

CLASE

Comparativa con los sistemas gestores relacionales

- Los sistemas XML nativos respetan el orden y ofrecen herramientas para tratarlos.
- Soportan elementos, atributos, comentarios y muchos otros aspectos XML, incluso la posibilidad de optionalidad de elementos de la estructura.
- Hay nuevos tipos de aplicaciones que requieren la extensibilidad de datos que ofrece XML.

Estrategias de almacenamiento

- En general, todos los sistemas XML nativos tienen como principal misión el almacenamiento y la gestión de documentos XML.



Características

- XML permite asignar una estructura a documentos XML mediante esquemas XML o DTD.
- Permiten almacenar y recuperar documentos acorde a la especificación XML.
- Como sistema gestor, cualquier sistema XML nativo debe soportar concurrencia y seguridad.
- Los sistemas XML nativos deben dar soporte a toda la tecnología asociada XML.

CLASE

Colecciones y documentos

- Una colección es un conjunto de documentos, de modo que forma una estructura de árbol donde cada documento pertenece a una única colección.



Ejemplo de colección

```
<!-- Base de datos de libros en Castellano -->
<Libros>
  <Libro>
    <Autor>Nikolai Gogol</Autor>
    <Titulo>El Capote</Titulo>
  </Libro>
  <Libro>
    <Autor>Gonzalo Giner</Autor>
    <Titulo>El Sanador de Caballos</Titulo>
  </Libro>
  <Libro>
    <Autor>Umberto Eco</Autor>
    <Titulo>El Nombre de la Rosa</Titulo>
  </Libro>
</Libros>
```

Ejemplo de documento

Librerías de acceso a datos XML

- Las librerías ampliamente utilizadas en desarrollos profesionales que permiten acceder desde Java a sistemas XML son:
 - XML-DB
 - XQJ



Librerías de acceso a datos XML

- Una diferencia importante entre XQJ y XML:DB es que el primero no se preocupa por las colecciones.
- Al igual que en una BBDD relacional, XQJ establece una conexión y luego accederá con un lenguaje.
 - En los sistemas relacionales, es SQL, en sistemas XQJ, es XQuery.

CLASE

Contexto en el que se usan las APIs

- XQJ:
 - XQJapi.jar, XQJ2-0.0.1.jar, eXist-XQJ-1.0.1.jar, eXist-XQJ-exemples.jar
- XML:DB:
 - Librerías localizadas en instalación de eXist.
 - Librerías localizadas en instalación de eXist/lib/core

eXist

- eXist es un sistema gestor nativo de XML de código libre.
- Los documentos se almacenan en colecciones y cada documento está en una colección.
- Además del entorno para poder usar las API anteriores, también tenemos que instalar el sistema eXist.

CLASE

eXist

- En el proceso de instalación, solicitará la versión del JDK a utilizar.
- La instalación también pedirá el nombre de usuario y la contraseña (admin/admin).
- También tenemos que instalar el servicio eXist.
 - Una vez instalado, arrancar el servicio eXist manualmente para poder acceder con la aplicación cliente.

CLASE

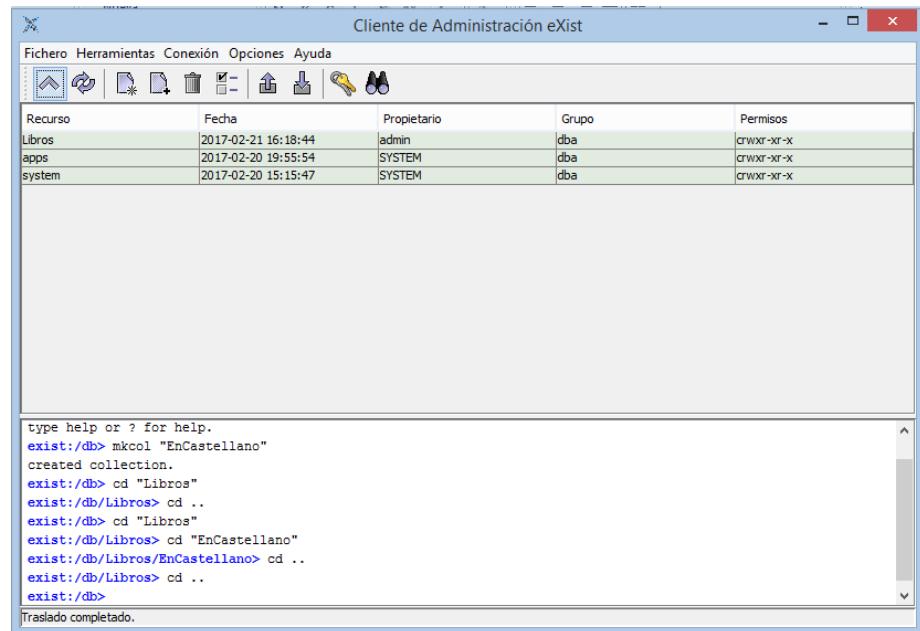
eXist

- El cliente por defecto que se instala es eXist Client Shell, que permite gestionar las colecciones y los recursos de la BBDD a las que tiene acceso el usuario registrado.



eXist

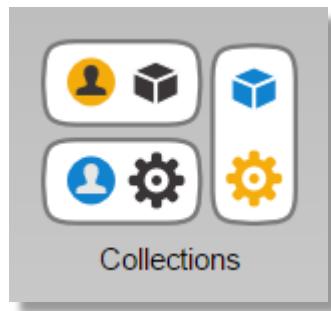
- <http://exist-db.org>
- <http://localhost:8080/exist>



CLASE

eXist

- Para crear una colección:



Collection Browser

Name	Permissions	Owner	Group	Last-modified
Libros	crwxr-xr-x	admin	dba	February 21 2017 16:18:44
apps	crwxr-xr-x	SYSTEM	dba	February 20 2017 19:55:54
system	crwxr-xr-x	SYSTEM	dba	February 20 2017 15:15:47

Cliente de Administración eXist

Fichero Herramientas Conexión Opciones Ayuda

Recurso	Fecha	Propietario	Grupo	Permisos
Libros	2017-02-21 16:18:44	admin	dba	crwxr-xr-x
apps	2017-02-20 19:55:54	SYSTEM	dba	crwxr-xr-x
system	2017-02-20 15:15:47	SYSTEM	dba	crwxr-xr-x

eXist

- Haciendo doble click en una colección, podemos añadir colecciones internas o recursos:

The screenshot shows the 'Collection Browser' window with the title 'Collection Browser'. At the top, there is a toolbar with various icons. Below the toolbar, the path '/db/Libros/EnIngles' is displayed in a text input field, which is highlighted with a red box. The main area is a table with the following data:

Name	Permissions	Owner	Group	Last-modified
..	crwxr-xr-x	admin	dba	February 21 2017 16:18:44
Tecnicos	crwxr-xr-x	admin	dba	February 21 2017 16:22:41
Libros_Ingles.xml	-rw-r--r--	admin	dba	February 21 2017 16:22:48

Conexión con XML:DB

- Para la conexión con XML:DB se necesitan los siguientes elementos:
 - El driver: para eXist el *driver="org.exist.xmldb.DatabaseImpl"*.
 - Se carga el driver en memoria:
c1 = Class.forName(driver);
 - Se crea una instancia de la BBDD y se guarda en una variable tipo Database:
database = (Database) c1.newInstance();

Conexión con XML:DB

- La instancia de la BBDD es necesario registrarla:
DatabaseManager.registerDatabase(database);
- Una vez registrada, ya podemos acceder a la colección deseada.
- Una ruta de la conexión se especifica mediante una URI y la ruta de la colección.

Conexión con XML:DB

```
protected static String driver = "org.exist.xmldb.DatabaseImpl";
public static String URI = "xmldb:exist://localhost:8080/exist/xmlrpc";
private Database database;
private String usuario = "admin";
private String usuarioPwd = "password";
private String collection = "/db/Libros/EnIngles";

Class cl = Class.forName(driver);
//Se crea un objeto Database
database = (Database) cl.newInstance();
DatabaseManager.registerDatabase(database);
//Ahora se obtiene la colección (URI + collection)
//con el usuario y password que tiene acceso a ella.
Collection col = DatabaseManager.getCollection(URI + collection, usuario,
    usuarioPwd);
if (col.getResourceCount ()== 0) {
    //si la colección no tiene recursos no podrá devolver ninguno
    System.out.println("La colección no tiene recursos..."
        + "No puede devolver ninguno [FIN]");
}
```

Conexión con XQJ

- Los pasos necesarios son:
 - Crear una fuente de datos (datasource):

```
XQDataSource xqs = new EXistXQDataSource();
```
 - Establecer las propiedades de la conexión:

```
xqs.setProperty("serverName", "localhost");  
xqs.setProperty("port", "8080");
```
 - Obtener una conexión:

```
XQConnection con = xqs.getConnection("admin", "admin");
```
- Cerrar la conexión:

```
con.close();
```

Conexión con XQJ

```
XQConnection conn;  
XQDataSource xqs;  
//Se crea el DataSource (origen de datos)  
xqs = new ExistXQDataSource();  
//Se establecen las propiedades de la conexión  
xqs.setProperty("serverName", "localhost");  
xqs.setProperty("port", "8080");  
//Se obtiene la conexión  
conn = xqs.getConnection("admin", "password");  
//Se cierra la conexión  
conn.close();
```

Creación y borrado de recursos

- XML:DB ofrece un nivel de abstracción menor que XQJ.
- XQJ se centra únicamente en ejecutar consultas XQuery obviando la estructura de recursos (documentos) y colecciones que subyace en los sistemas XML nativos.
- XML:DB da la posibilidad de crear o eliminar colecciones y recursos.

Accediendo a recursos con XML:DB

- XML:DB llama a los documentos (XML o no-XML) recursos (resources).
- Resource: es una clase contenedora de los datos almacenados en la BBDD.
- XMLResource: extiende a la anterior clase y proporciona acceso a recursos solo de tipo XML almacenados en la BBDD.

CLASE

Accediendo a recursos con XML:DB

```
    } else {
        System.out.println("...La colección no es nula...");
        Resource res = null;
        String nombre_recurso = "Libros_Ingles.xml";
        // Se obtiene el recurso "Libros_Ingles.xml"
        res = (Resource) col.getResource(nombre_recurso);
        System.out.println("De la colección saca el recurso que tiene la "
            + "variable nombre_recurso:" + nombre_recurso);
        // Se usa un molde para convertir res de tipo Resource a XMLResource
        XMLResource xmlres = (XMLResource) res;
        // Se saca el contenido del recurso y se muestra en pantalla
        System.out.println("La salida es:\n" + xmlres.getContent());
    }
```

CLASE

Creando recursos con XML:DB

- En una colección podemos añadir nuevos recursos XML y no-XML. Para ello utilizamos las siguientes clases y métodos:
 - Collection:
Representa una colección de recursos almacenados en la BBDD XML.

CLASE

Creando recursos con XML:DB

- Métodos:
 - *storeResource()*
 - *removeResource()*
 - *listResource()*
 - *getResourceCount()*
 - *createResource()*

Creando recursos con XML:DB

```
public void asignarRecursoBD(Collection contexto, File archivo)
    throws ExcepcionGestorBD {
try {
    // Crea un recurso vacío
    Resource nuevoRecurso = contexto.createResource(archivo.getName(),
        "XMLResource");
    // Asigna contenido del archivo al nuevo recurso vacío
    nuevoRecurso.setContent(archivo);
    // Almacena el recurso en la colección
    contexto.storeResource(nuevoRecurso);
} catch (XMLDBException e) {
    throw new ExcepcionGestorBD("error XMLDB :" + e.getMessage());
}
}
```

CLASE

Borrado de recursos con XML:DB

- Los métodos más destacados para el borrado de recursos son:
 - *removeResource(Resource re)*
Elimina el recurso res de la colección
 - *getResource(String id)*
Recupera un recurso a través de su id.

CLASE

Borrado de recursos con XML:DB

```
public void borrarRecurso(Collection contexto, File archivo)
    throws ExcepcionGestorBD {
try {
    Resource res = (Resource) col.getResource(archivo.getName());
    contexto.removeResource(res);
} catch (XMLDBException e) {
    throw new ExcepcionGestorBD("error XMLDB :" + e.getMessage());
}
}
```

CLASE

Modificación de contenidos XML

- Es necesario poder consultar y actualizar los datos.
- Para realizar actualizaciones existen alternativas:
 - El uso de DOM o SAX.
 - Xupdate.
 - Xquery y su extensión XQueryUpdateFacility3.0.
 - Lenguajes declarativos propietarios que ofertan los diferentes sistemas XML nativos.

CLASE

Actividad 04

Base de datos con XND.

- Fecha de entrega: 19/04/2022
- Proyecto creado en NetBeans
comprimido en .zip

CLASE



Linkia FP

Formación Profesional Oficial a Distancia