



# Bases de datos Nativas XML

(Repaso de XML)



# Bases De Datos Nativas Xml

- **Repaso XML**
  - ◆ Qué es
  - ◆ Para qué sirve
  - ◆ XML vs HTML
  - ◆ Qué no hace
  - ◆ Ventajas
  - ◆ Árbol
  - ◆ Sintaxis, elementos y atributos
  - ◆ Analizadores: DOM y SAX
  - ◆ Serializador
  - ◆ SOAP



# Repaso XML

- Qué es XML
  - ◆ XML (eXtensible Markup Lenguaje)
  - ◆ Estándar del W3C
  - ◆ Metalenguaje de definición de etiquetas estructuradas
  - ◆ Define reglas semánticas → Organizar la información

# Repaso XML

Dónde se encuentra el XML

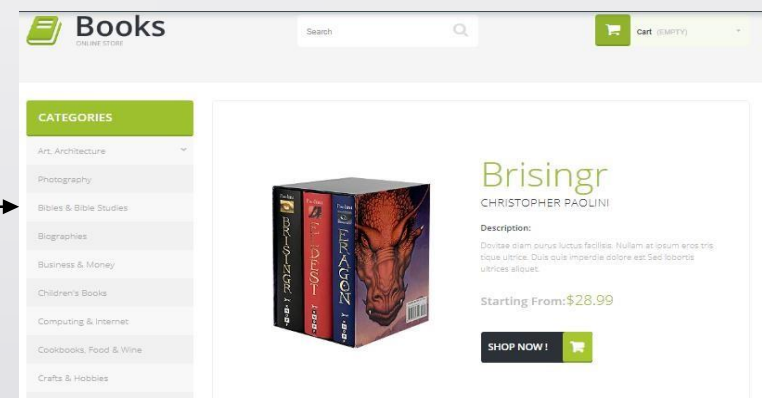
SEÑAL DE  
ENTRADA

SISTEMA  
Procesado

SEÑAL DE  
SALIDA

```
1 <bookstore>
2   <book category="infantiles">
3     <title>Harry Potter</title>
4     <author>J. K. Rowling</author>
5     <year>2005</year>
6     <price>25.00</price>
7     <quantity>3</quantity>
8   </book>
9   <book category="sistemas de televisión">
10    <title>Sistemas Audiovisuales</title>
11    <author>Francisco Tarres Ruiz</author>
12    <year>2000</year>
13    <price>20.00</price>
14    <quantity>3</quantity>
15  </book>
16 </bookstore>
```

ORDEN	TITULOS MAS LEIDOS	AUTORES
1	EL CODIGO DAVINCI	DAN BROWN
2	ANGELES Y DEMONIOS	DAN BROWN
3	LA SOMBRA DEL VIENTO	CARLOS RUIZ ZAFÓN
4	LOS PILARES DE LA TIERRA	KEN FOLLET
5	EL QUIJOTE	MIGUEL DE CERVANTES
6	LA CONSPIRACION	DAN BROWN
7	LA HERMANDAD DE LA SABANA SANTA	JULIANAVARRO
8	EL SEÑOR DE LOS ANELLOS	J.R.R. TOLKIEN
9	EL OCHO	KATHERINE NEVILLE
10	EL ULTIMO CATÓN	MATILDE ASENSI
11	LA BIBLIA DE BARRO	JULIANAVARRO
12	HARRY POTTER	J.K. ROWLING
13	DEJAME QUE TE CUENTE	JORGE BUCAY
14	EL ULTIMO MEROVINGIO	JIM HOGAN



# Para qué sirve

✓ Es texto plano

- ▶ Almacenar y transportar datos
- ▶ Una máquina y un humano pueden leerlo
- ▶ Distribuir datos a través de Internet

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Irache</to>
  <from>Agurtxane</from>
  <heading>Recordatorio</heading>
  <body>No te olvides que quedamos este fin de semana</body>
</note>
```



## XML vs HTML

- ◆ Objetivos diferentes
- ◆ XML: Transportar datos
- ◆ HTML: mostrar datos, dar formato
- ◆ Usan lenguaje de etiquetas diferentes.
- ◆ XML no utiliza etiquetas predefinidas. Podemos añadir etiquetas inventadas
- ◆ HTML usa etiquetas definidas y conocidas <p> <h1> <table>...



## Qué no hace

- ▶ Sólo sirve para presentar datos/información
- ▶ Es un lenguaje que envuelve en etiquetas los datos/información

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Irache</to>
  <from>Agurtxane</from>
  <heading>Recordatorio</heading>
  <body>No te olvides que quedamos este fin de semana</body>
</note>
```

- ▶ Alguien tiene que hacer software para enviar, recibir, almacenar, procesar o mostrar el texto.



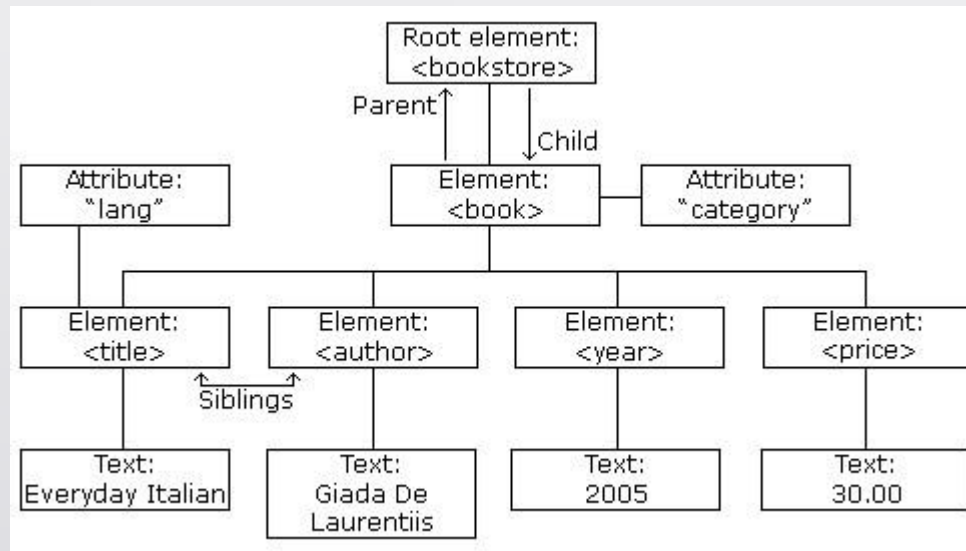
## Ventajas XML

- ☐ Bien formado
- ☐ Extensible
- ☐ Facilidad de conversión
- ☐ Fácil de leer
- ☐ Auto descriptivo
- ☐ Intercambiable
- ☐ Lectura/Interpretación -> Parser



# Árbol XML

- ◆ Estructura en árbol.
- ◆ Empieza en la raíz y se ramifica en hojas.



```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="infantiles">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>25.00</price>
    <quantity>8</quantity>
  </book>
  <book category="sistemas de televisión">
    <title>Sistemas Audiovisuales</title>
    <author>Francesc Tarrés Ruiz</author>
    <year>2000</year>
    <price>20.00</price>
    <quantity>3</quantity>
  </book>
</bookstore>
```

## Sintaxis XML

- Empieza por el Elemento Raíz (padre de todos los eltos.)

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Irache</to>
  <from>Agurtxane</from>
  <heading>Recordatorio</heading>
  <body>No te olvides que quedamos este fin de semana</body>
</note>
```

## Sintaxis XML

- **Atributos** - entre comillas

```
<?xml version="1.0" encoding="UTF-8"?>
<note date="05/02/2021">
  <to>Irache</to>
  <from>Agurtxane</from>
  <heading>Recordatorio</heading>
  <body>No te olvides que quedamos este fin de semana</body>
</note>
```



## Elementos XML

- Documento XML - tiene eltos. XML
- Pueden tener:
  - Texto
  - Atributos
  - Otros eltos.
  - Mezcla
- Eltos vs. Atributos

```
<person gender="female">
  <firstname>Ana</firstname>
  <lastname>Garcia</lastname>
</person>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="infantiles">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>25.00</price>
    <quantity>8</quantity>
  </book>
  <book category="sistemas de televisión">
    <title>Sistemas Audiovisuales</title>
    <author>Francesc Tarrés Ruíz</author>
    <year>2000</year>
    <price>20.00</price>
    <quantity>3</quantity>
  </book>
</bookstore>
```



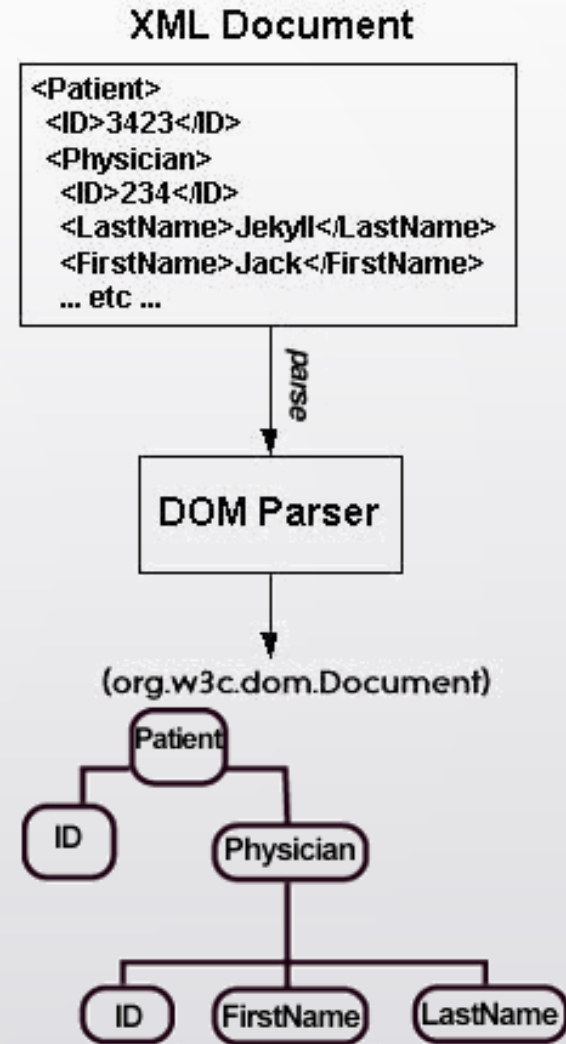
## XML Parser (Analizadores)

- Biblioteca o software que proporciona una interfaz para que las aplicaciones cliente trabajen con documentos XML.
- Comprueba el formato adecuado XML
- Valida los documentos XML
- La mayoría de navegadores tienen un XML parser integrado.
- Los más conocidos son:
  - DOM: en forma de árbol
  - SAX: forma secuencial



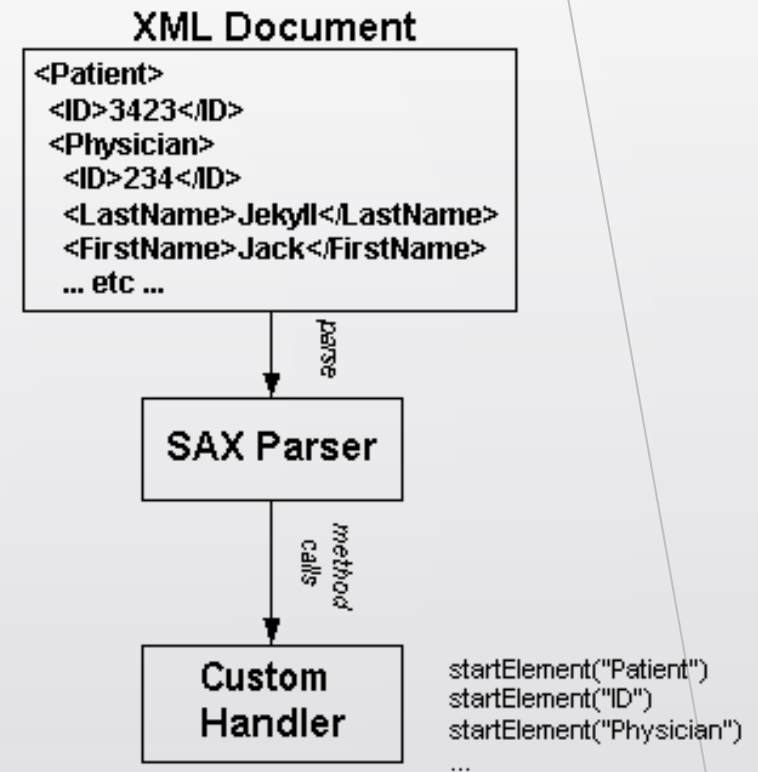
## XML Parser: DOM

- ◆ Acceso a los elementos del XML a través del DOM
- ◆ Modelo de objetos estándar
- ◆ Interfaz de programación estándar
- ◆ Independiente de la plataforma
- ◆ Estándar del W3C



## XML Parser: SAX

- ✓ Acceso a los elementos del XML a través del SAX
- ✓ Lee el documento de elemento en elemento
- ✓ Consume menos memoria



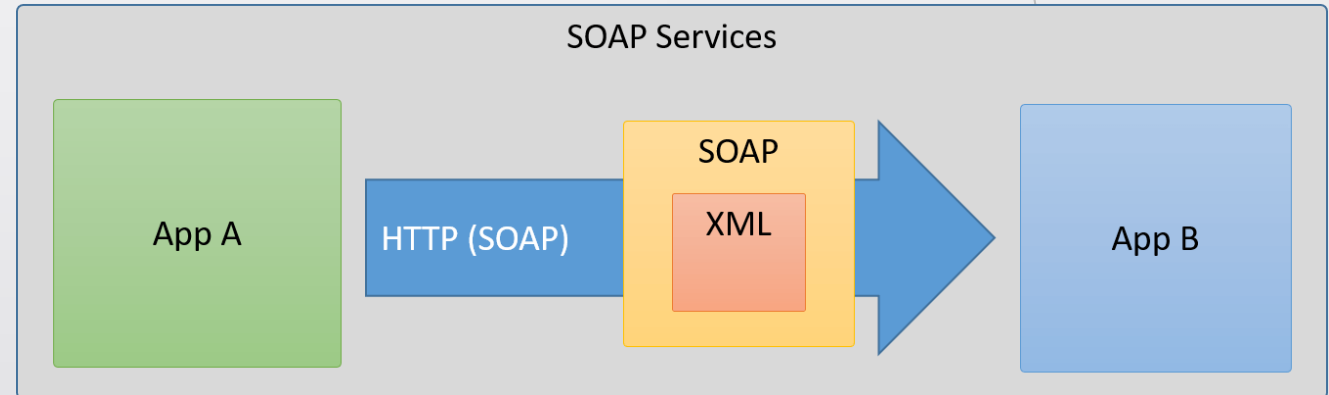


## XML serializer

- ◆ Convierte objetos Java en documentos XML.
- ◆ Existen FrameWorks específicos que pueden cargarse desde Apache Maven.
- ◆ Intercambio de objetos entre aplicaciones o operaciones CRUD en bases de datos.

## Ejemplo particular: SOAP

- Simple Object Access Protocol
  - Protocolo de comunicación entre aplicaciones
  - Define un formato para enviar y recibir mensajes
  - Independiente de la plataforma
  - Basado en XML
  - Recomendación del W3C
  - Funciona sobre HTTP, FTP...
  - Poca flexibilidad (aparece REST)





# Bases de datos Nativas

## XML (Introducción)





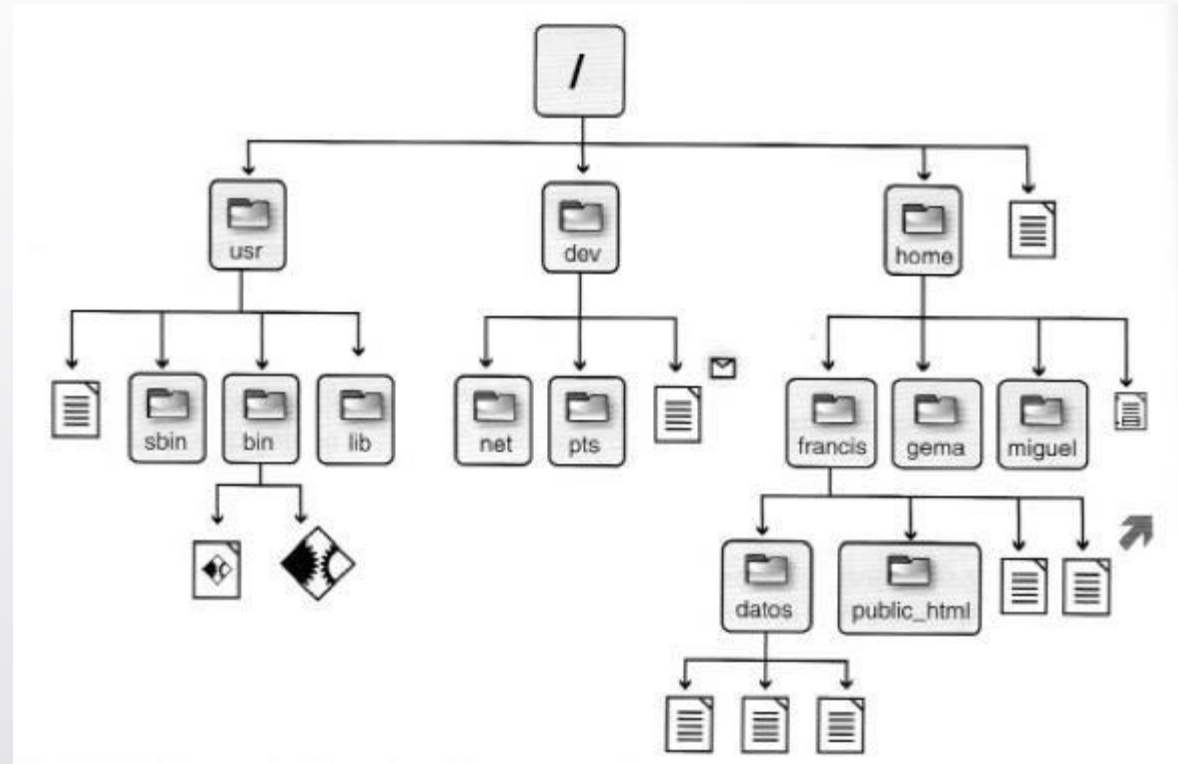
# Bases de datos Nativas XML (Introducción)

- Índice

1. Cómo se almacenan los XML
2. Bases de datos Nativas XML
3. Definición
4. Ventajas
5. Desventajas
6. Implementaciones comerciales

## ¿Cómo almacenar XML?

- Semejanza Sistemas de ficheros





## Bases de datos nativas XML (Native XmlDatabase)

- ◆ No poseen campos
- ◆ No almacenan datos
- ◆ Almacenan documentos XML
- ◆ Unidad mínima de información: Documento XML



## Bases de datos nativas XML (Native XmlDatabase)

- Definición
  - ◆ Define un modelo de datos XML
  - ◆ Utiliza el documento como unidad mínima de almacenamiento
  - ◆ Puede usar cualquier estrategia de almacenamiento
  - ◆ Permite tecnologías de consulta y transformación propias



## Bases de datos nativas XML (Native XmlDatabase)

- Ventajas
  - ◆ Acceso y almacenamiento directo en XML
  - ◆ Motor de búsqueda de alto rendimiento
  - ◆ Sencillo añadir nuevos XML al repositorio
  - ◆ Almacena datos heterogéneos
- Desventajas
  - ◆ Difícil indexado
  - ◆ No admiten agregación
  - ◆ Formación de nuevas estructuras complicada y lenta



## Implementaciones comerciales

<ul style="list-style-type: none"><li>◆ <b>Licencia comercial</b><ul style="list-style-type: none"><li>◆ eXcelon XIS</li><li>◆ GoXML DB</li><li>◆ Infonyte-DB</li><li>◆ Tamino</li></ul></li></ul>	<ul style="list-style-type: none"><li>◆ <b>Código abierto</b><ul style="list-style-type: none"><li>◆ dbXML</li><li>◆ eXist</li><li>◆ Xindice</li></ul></li></ul>	<ul style="list-style-type: none"><li>◆ <b>Investigación:</b><ul style="list-style-type: none"><li>◆ Lore</li><li>◆ Natix</li><li>◆ PDOM</li><li>◆ TIMBER</li></ul></li></ul>
--	--	---



## XML (eXist-DB)

- Índice

1. eXist DB
2. Instalación
3. Configuración
4. Cliente de administración
5. Colecciones
6. Subir documentos XML

## eXist-DB

- ☐ Base de datos nativa XML Open Source
- ☐ Desarrollada en Java ☐
- ☐ **Funcionamiento:**
  - ☐ Cliente/Servidor
  - ☐ Independiente
- ☐ Integrada en la aplicación
- ☐ **Lenguaje de consultas**
  - ☐ XPath
  - ☐ XQuery





## eXist-DB

- ☐ Base de datos nativa XML Open Source
- ☐ Desarrollada en Java
- ☐ **Funcionamiento:**
  - ☐ Servidor autónomo atendiendo a llamadas de XML-RPC, **WebDAV** y REST
- ☐ Insertado dentro de una aplicación JAVA
- ☐ En un servidor J2EE, dando servicios XML-RPC, SOAP y **WebDAV**



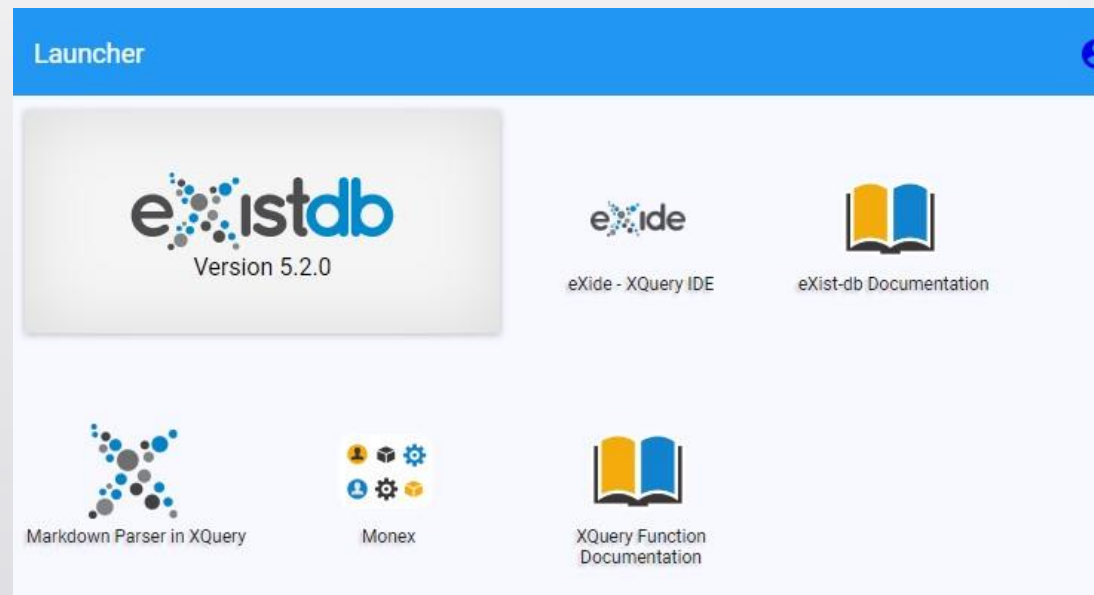
# eXist-db

- Instalaciones
  - Docker
    - <https://hub.docker.com/r/existdb/existdb/tags/>
  - Web Oficial: descargar e instalar <http://www.exist-db.org>
  - java -jar exist-installer-6.0.1.jar
    - BD
      - (admin, "")




## Instalación eXist-DB

- ◆ Por defecto se instala como proceso en el puerto 8080.
- ◆ Ir al explorador e introducir `http://myIP:8080`,
- ◆ Por ejemplo, si mi IP es 192.168.1.150, introducir: `http://192.168.1.150:8080`




# eXist-db


Mejor coincidencia

 **eXist-db Database**  
Aplicación


Aplicaciones

 **Uninstall eXist-db** >


Buscar en Internet


 **exi - Ver resultados web** >


Documentos (1+)



**eXist-db Database**  
Aplicación

 Abrir

 Abrir ubicación de archivo

 Desanclar de Inicio

eXist-db Configuration

It seems you are starting eXist-db for the first time. Please configure your memory settings below.

**After startup, use dashboard or Java client to set a password for admin (empty by default).**

Java Memory

Memory usage: 78 free/4068 max mb

Min Memory  Memory settings only become effective after restart and only apply when eXist-db is started via the system tray launcher.

Max Memory

Caches

General Cache  Total cache size should not exceed 1/3 of max memory unless you have more than 2GB available. These sizes are in megabytes.

Collection Cache

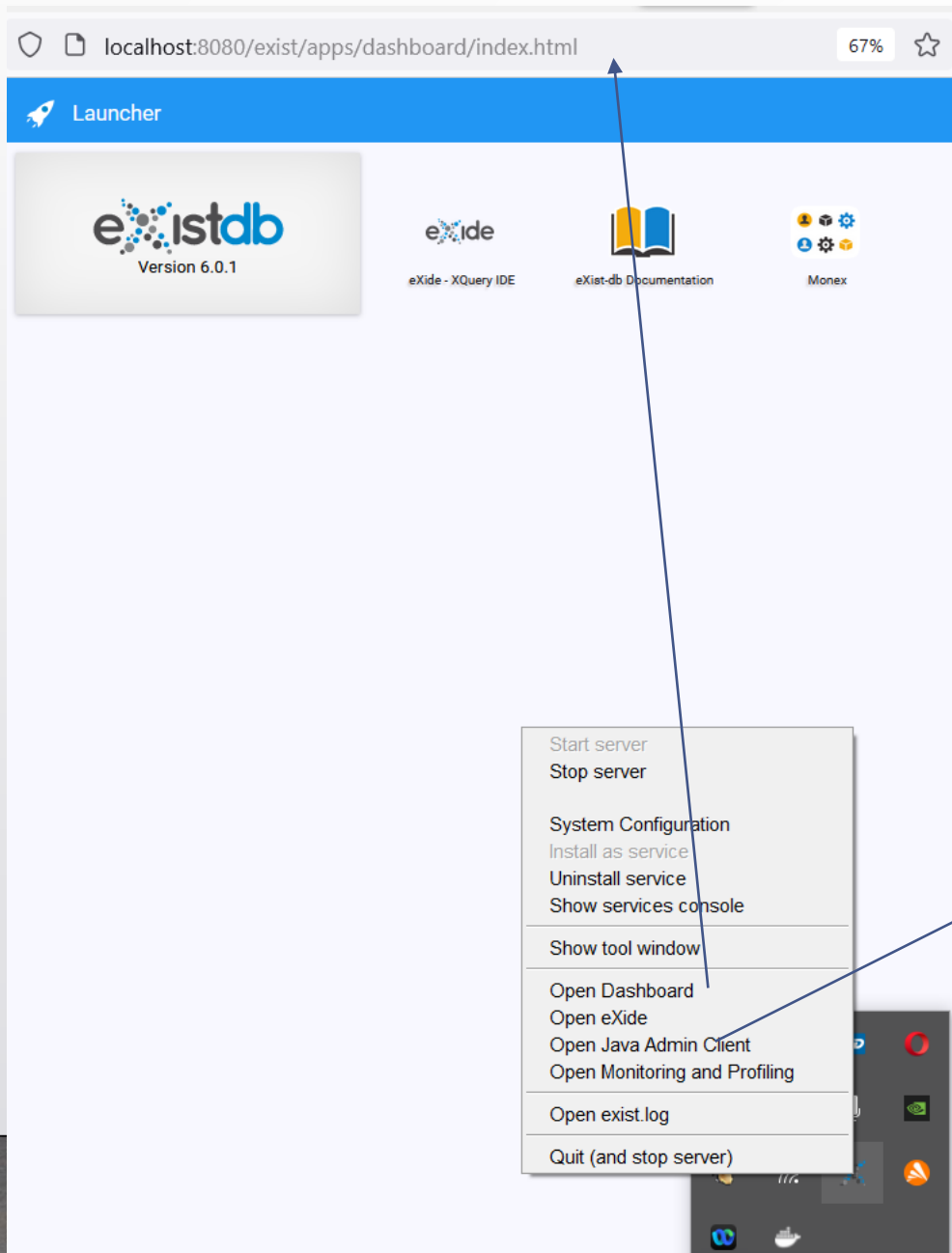
Data Directory

Changing the data directory will create an empty database in the new location (unless there's already data in it).

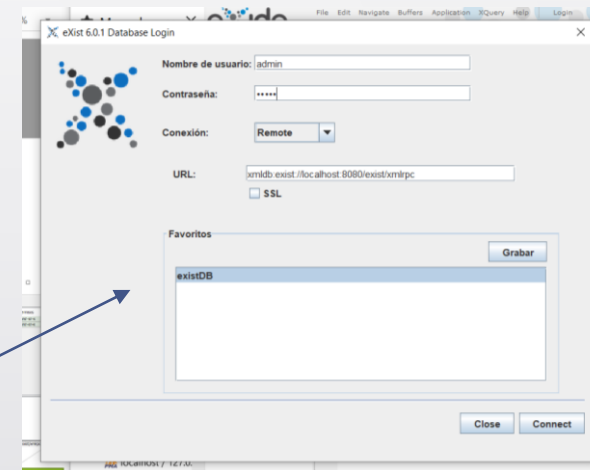
Jetty Ports

HTTP Port  Set the ports used by the integrated web server. Please make sure those ports are not used by other processes.

SSL Port



# eXist-db



# eXist-db

Cliente de Administración eXist

Fichero Herramientas Conexión Opciones Ayuda

🏠 🔄 📄 🗑️ 📁 📥 🔑 🔍

Recurso	Fecha	Propietario	Grupo	Permisos
ColeccionesXML	2022-03-19 21:4...	admin	dba	crwxr-xr-x
apps	2022-03-19 21:3...	SYSTEM	dba	crwxr-xr-x
system	2022-03-19 21:3...	SYSTEM	dba	crwxr-xr-x

type help or ? for help.  
exist:/db>

Seleccionar los ficheros o directorios a almacenar

Look in: ColeccionesXML

Elementos re...

- BDProductosXML
- ColeccionGimnasio
- ColeccionPruebas
- ColeccionVentas

Escritorio

Documentos

Este equipo

Red

File name: curso\_2021\_2022\AADD\UD5\_BDXML\ProyectoJAVA\ColeccionesXML

Files of type: All Files

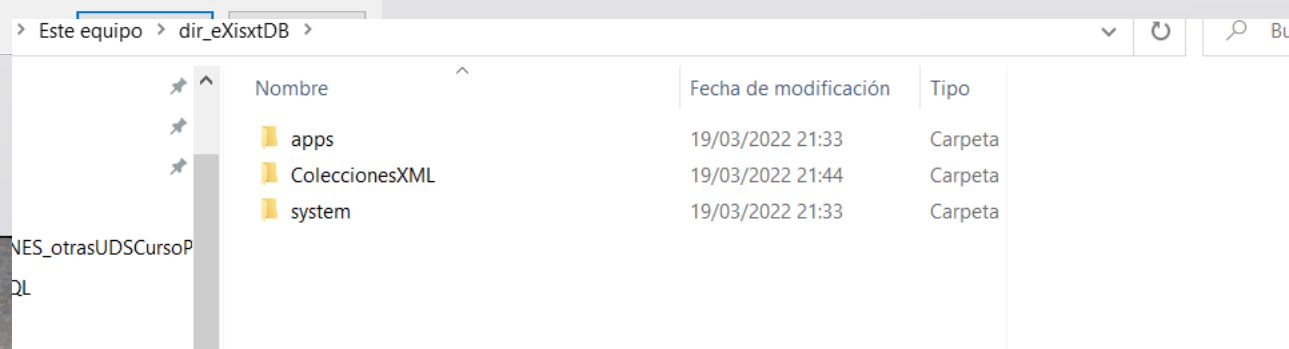
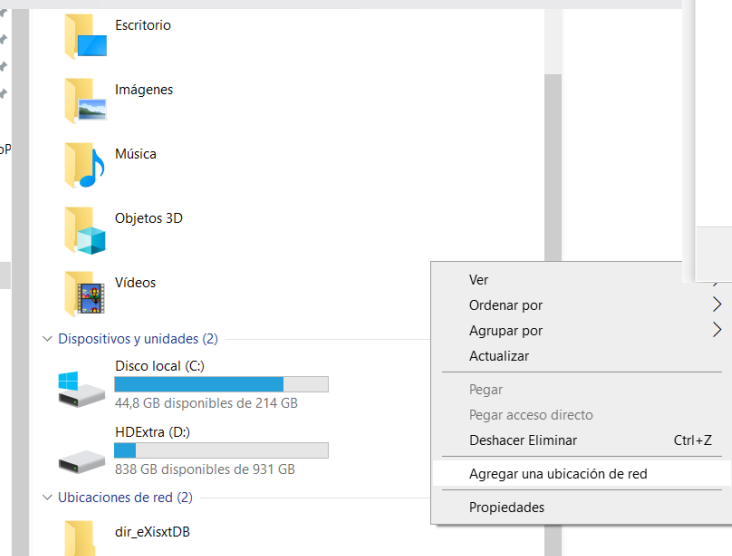
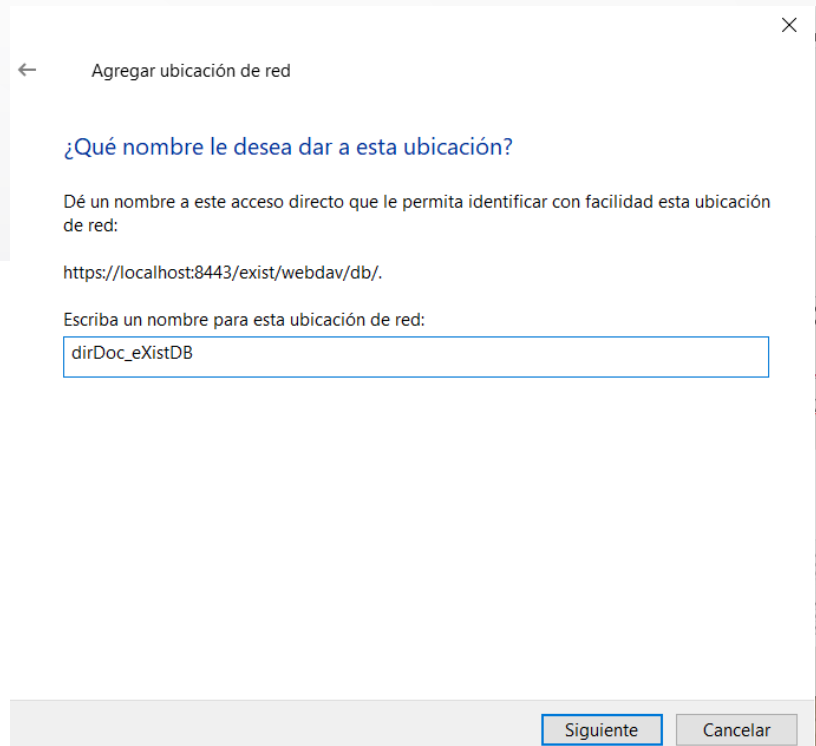
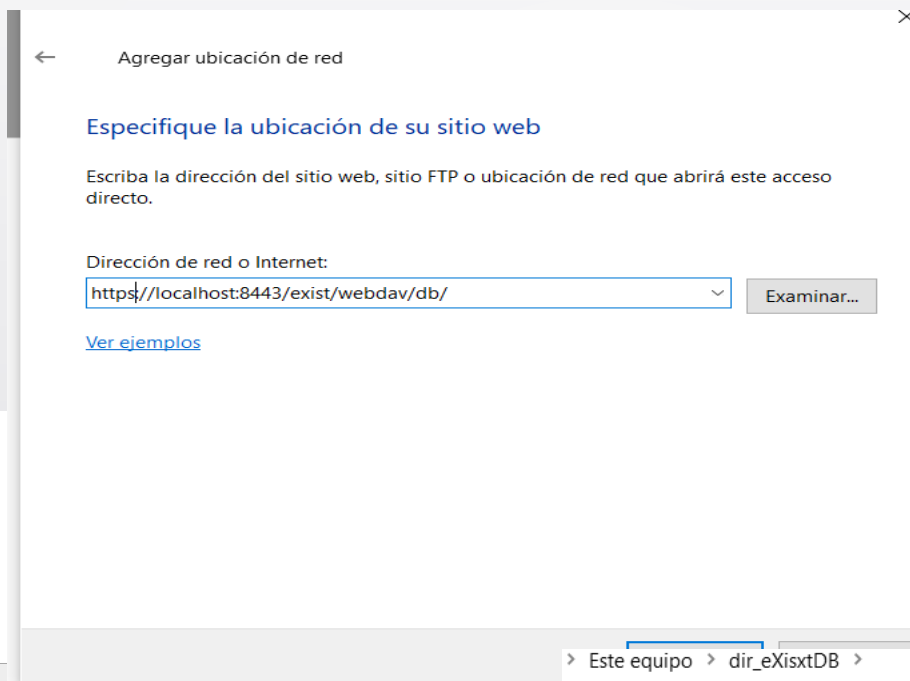
Seleccionar los ficheros o directorios a almacenar

Cancel

- WebDAV

- `https://localhost:8443/exist/webdav/db`

eXist-db





# eXist-db

- Colecciones
  - ◆ Uso de colecciones
  - ◆ Agrupaciones de documentos XML, binarios y otras colecciones.
  - ◆ Colección similar a directorio en sistema de ficheros.
  - ◆ La colección padre es /db/
- Subir NuevoDoc XML
  - ◆ PRACTICA - CREAR Y SUBIR FICHEROS A BD
  - ◆ Usar el Notepad++ para crear nuestros propios documentos XML

A screenshot of a code editor window titled 'alumnos.xml'. The window displays an XML document with the following content:

```
1 <?xml version="1.0"?>
2 <listadealumnos>
3   <alumno>
4     <nombre>Juan</nombre>
5     <edad>19</edad>
6   </alumno>
7   <alumno>
8     <nombre>Maria</nombre>
9     <edad>20</edad>
10  </alumno>
11 </listadealumnos>
```





# Índice

- Estándares
  1. Xpath
  2. XQuery



## XPath

- ☐ Lenguaje que permite seleccionar nodos de un documento XML
- ☐ Calcular valores
- ☐ Selecciona partes del documento a partir de:
- ☐ Nodo raíz
- ☐ Nodos elemento
- ☐ Nodos atributo
- ☐ Nodos comentario
- ☐ Nodos espacio de nombres
- ☐ Nodos de instrucción de proceso

### Realización ejercicios prácticos

- Consultas
- Conexión/Uso desde Java



## XQuery

- Lee datos de uno o más documentos XML
- Contiene a XPath
- Mucho más versátil
- Siguen la norma **FLWOR** (flower): For, Let, Where, Order y Return
  - for <variable> in <expresión XPath>
  - let <variables vinculadas>
  - where <condición XPath>
  - order by <expresion>
  - return <expresión de salida>

### Realización ejercicios prácticos

- Consultas
- Conexión/Uso desde Java



Usando Java

ACCESO A DATOS

Bases de datos Nativas

XML (XML:DB, XQJ)



# Índice

## 2 APIs acceso a eXist DB

- XML:DB
  1. Maven/Ant
  2. Driver
  3. Collection
  4. Servicios
  5. Action (CRUD)
- XQJ



## XML:DB - Creación del driver

- Añadir al fichero **pom.xml**:

```
<dependency>
```

```
    <groupId>org.exist-db</groupId>
```

```
    <artifactId>exist-core</artifactId>
```

```
    <version>5.2.0</version>
```

```
</dependency>
```

```
<dependency>
```

```
    <groupId>net.sf.xmlldb-org</groupId>
```

```
    <artifactId>xmlldb-api</artifactId>
```

```
    <version>1.7.0</version>
```

```
</dependency>
```



## XML:DB - Librerías

- *exist.jar,*
- *exist-optional.jar,*
- *xmlldb.jar,*
- *xml-apis-1.3.04.jar,*
- *xmlrpc-client-3.1.1.jar,*
- *xmlrpc-common-3.1.1.jar*
- *y log4j-1.2.15.jar*

## XML:DB

```
// Declaración del driver de XML:DB para eXist
```

```
String driver = "org.exist.xmldb.DatabaseImpl";
```

```
// Inicializar el driver
```

- `Class cl = Class.forName(driver); // cargar el driver`
- `Database database = (Database) cl.newInstance(); // Instanciamos la BD (depricated)`  
`cl.getDeclaredConstructor().newInstance()`

```
DatabaseManager.registerDatabase(database); // registro de la BD
```

## XML:DB - Collection

Ejemplo:

```
// Creamos una colección con un import del tipo:
```

```
// import org.xmldb.api.base.Collection;
```

```
// Servirá para conectarnos a la base de datos eXist-DB
```

```
// Indicamos los datos básicos de conexión (uri, user, password):
```

```
String uri = "xmldb:exist://localhost:8080/exist/xmlrpc/db/  
ColeccionesXML/ColeccionPruebas ";
```

```
String user = "admin"; String pass = "";
```

```
Collection col = (Collection) DatabaseManager.getCollection(uri, user, pass);
```



## XML:DB - Servicios

Para consultar el repositorio (la base de datos), debemos usar:

1. `XPathQueryService`
2. `XQueryService` de eXist
  - ☐ Internamente, eXist DB no distingue entre expresiones XPath y XQuery.
  - ☐ Por lo tanto, `XPathQueryService` y `XQueryService` son lo mismo pero la segunda aporta algunos métodos adicionales



## XML:DB - Servicios (XPathQueryService)

// Creamos el servicio:

```
XPathQueryService service = (XPathQueryService) col.getService("XPathQueryService", "1.0");
```

// creamos la consulta

```
String strAction = "for $emp in /EMPLEADOS/EMP_ROW[DEPT_NO = 10] return $emp";
```

// Para ejecutar la consulta se llama al método: `service.query(xpath)`. Este método devuelve un objeto de la clase 'ResourceSet' que contiene el resultado de la consulta.

```
ResourceSet result = service.query(strAction);
```

// Vamos avanzando en cada uno de los resultados con ayuda del método `.getIterator()` sobre nuestro resultado.

```
ResourceIterator i = result.getIterator();
```

```
i.hasMoreResources() // Ver si hay algo:
```

// Mostrar la respuesta

```
Resource r = (Resource) i.nextResource();
```

```
System.out.println((String) r.getContent());
```

## XML:DB - Action (Create)

◆ Ejemplo:

update insert

<DEP\_ROW>

<DEPT\_NO>55</DEPT\_NO>

<DNOMBRE>ingenieria</DNOMBRE>

<LOC>sanxenxo</LOC>

</DEP\_ROW>

into /departamentos

```
<departamentos>
  <TITULO>DATOS DE LA TABLA DEPART</TITULO>
  <DEP_ROW>
    <DEPT_NO>10</DEPT_NO>
    <DNOMBRE>CONTABILIDAD</DNOMBRE>
    <LOC>SEVILLA</LOC>
  </DEP_ROW>
  <DEP_ROW>
    <DEPT_NO>55</DEPT_NO>
    <DNOMBRE>ingenieria</DNOMBRE>
    <LOC>sanxenxo</LOC>
  </DEP_ROW>
</departamentos>
```

## XML:DB - Action (Delete)

### ◆ Ejemplo:

update delete /departamentos/DEP\_ROW[DEPT\_NO=55]

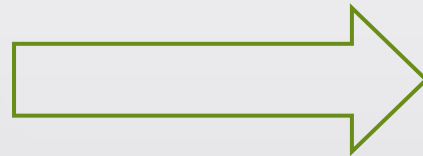
```
<departamentos>
  <TITULO>DATOS DE LA TABLA DEPART</TITULO>
  <DEP_ROW>
    <DEPT_NO>10</DEPT_NO>
    <DNOMBRE>CONTABILIDAD</DNOMBRE>
    <LOC>SEVILLA</LOC>
  </DEP_ROW>
  <DEP_ROW>
    <DEPT_NO>55</DEPT_NO>
    <DNOMBRE>ingenieria</DNOMBRE>
    <LOC>sanxenxo</LOC>
  </DEP_ROW>
</departamentos>
```

## XML:DB - Action(Update)

updatevalue

/departamentos/DEP\_ROW[DEPT\_NO=55]/DNOMBRE with 'Arquitectura'

```
<departamentos>
<TITULO>DATOS DE LA TABLA DEPART</TITULO>
<DEP_ROW>
  <DEPT_NO>10</DEPT_NO>
  <DNOMBRE>CONTABILIDAD</DNOMBRE>
  <LOC>SEVILLA</LOC>
</DEP_ROW>
<DEP_ROW>
  <DEPT_NO>55</DEPT_NO>
  <DNOMBRE>ingenieria</DNOMBRE>
  <LOC>sanxenxo</LOC>
</DEP_ROW>
</departamentos>
```



```
<departamentos>
<TITULO>DATOS DE LA TABLA DEPART</TITULO>
<DEP_ROW>
  <DEPT_NO>10</DEPT_NO>
  <DNOMBRE>CONTABILIDAD</DNOMBRE>
  <LOC>SEVILLA</LOC>
</DEP_ROW>
<DEP_ROW>
  <DEPT_NO>55</DEPT_NO>
  <DNOMBRE>Arquitectura</DNOMBRE>
  <LOC>Sanxenxo</LOC>
</DEP_ROW>
</departamentos>
```



## XML:DB - Action (Read All)

### ► Ejemplo:

```
for $dep in /departamentos return $dep
```

```
<departamentos>
<TITULO>DATOS DE LA TABLA DEPART</TITULO>
<DEP_ROW>
  <DEPT_NO>10</DEPT_NO>
  <DNOMBRE>CONTABILIDAD</DNOMBRE>
  <LOC>SEVILLA</LOC>
</DEP_ROW>
<DEP_ROW>
  <DEPT_NO>20</DEPT_NO>
  <DNOMBRE>INVESTIGACION</DNOMBRE>
  <LOC>MADRID</LOC>
</DEP_ROW>
<DEP_ROW>
  <DEPT_NO>30</DEPT_NO>
  <DNOMBRE>VENTAS</DNOMBRE>
  <LOC>BARCELONA</LOC>
</DEP_ROW>
<DEP_ROW>
  <DEPT_NO>40</DEPT_NO>
  <DNOMBRE>PRODUCCION</DNOMBRE>
  <LOC>BILBAO</LOC>
</DEP_ROW>
</departamentos>
```

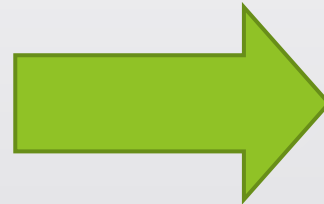


## XML:DB - Action (Búsqueda)

### ► Ejemplo:

```
for $dep in /departamentos/DEP_ROW[DEPT_NO=10] return $dep
```

```
<departamentos>  
<TITULO>DATOS DE LA TABLA DEPART</TITULO>  
<DEP_ROW>  
  <DEPT_NO>10</DEPT_NO>  
  <DNOMBRE>CONTABILIDAD</DNOMBRE>  
  <LOC>SEVILLA</LOC>  
</DEP_ROW>  
<DEP_ROW>  
  <DEPT_NO>20</DEPT_NO>  
  <DNOMBRE>INVESTIGACION</DNOMBRE>  
  <LOC>MADRID</LOC>  
</DEP_ROW>  
<DEP_ROW>  
  <DEPT_NO>30</DEPT_NO>  
  <DNOMBRE>VENTAS</DNOMBRE>  
  <LOC>BARCELONA</LOC>  
</DEP_ROW>  
<DEP_ROW>
```



```
<DEP_ROW>  
  <DEPT_NO>10</DEPT_NO>  
  <DNOMBRE>CONTABILIDAD</DNOMBRE>  
  <LOC>SEVILLA</LOC>  
</DEP_ROW>
```

# XML:DB

- Crear/Borrar una colección

▫En eXist-DB una colección es similar a una carpeta en un sistema de ficheros.

- 1.Disponer del Driver
- 2.Establecer la conexión (Collection)
- 3.Crear/borrar colección dentro de base de datos eXist-DB

▫Ejemplo:

```
CollectionManagementService mgtService =  
(CollectionManagementService)
```

```
col.getService("CollectionManagementService", "1.0");
```

```
// Crear nueva colección
```

```
mgtService.createCollection("nombreDeLaColeccionNueva");
```

```
// Eliminar colección existen
```

```
mgtService.removeCollection("nombreDeLaColeccionAEliminar");
```

- Subir/Borrar un archivo

▫Ejemplo:

```
// Subir un fichero que haya en el disco duro:
```

```
File file= new File("ARCHIVO_A_SUBIR.xml");
```

```
Resource resource = col.createResource(file.getName(),  
"XMLResource"); nuevoRecurso.setContent(file);  
col.storeResource(resource);
```

```
// Borrar un archivo de eXist-DB:
```

```
Resource recursoParaBorrar = col.getResource("ARCHIVO.XML");  
col.removeResource(recursoParaBorrar);
```

## XML:DB - Ejercicios

- Realiza una conexión con la BD para acceder al documento empleados.xml y obtener los empleados del departamento 10. (verempleados10())
- Implementar **vercolecciones()** y **verrecursosdelascolecciones()**
- Crear – **crearcoleccysubirarchivo(ruta)** o borrar – **borrarcoleccion(col)**

**borrarfichero (ruta,col)** una colección puede ser:

//Crear una nueva colección

```
CollectionManagementService mgtService = (CollectionManagementService)
coleccion.getService("CollectionManagementService", "1.0");
mgtService.createCollection("nuevaColeccion");
```

//Borrar una colección

```
CollectionManagementService mgtService = (CollectionManagementService)
coleccion.getService("CollectionManagementService", "1.0");
mgtService.removeCollection("nuevaColeccion");
```

## XML:DB - Ejercicios

- **añade un documento** nuevo a la colección, el documento es *NuevosDepartamentos.xml*, y se encuentra en la carpeta del proyecto Java. Se utiliza el paquete **java.io.File**, para declarar el fichero a subir a la BD, y el método ***createResource()*** para crear el recurso.
  - ***crearcoleccysubirarchivo***(String colecc)
- **borra el documento** anterior de la colección, comprobando si existe. Se utiliza el método
  - ***removeResource()***.
- **Actualizar el stock**
  - ***actualizaproductos()*** //Añadir 10 al stock
- Buscar, extraer y guardar un documento
  - ***bajardocumento()*** // Localizar un documento, extraerlo y guardarlo en disco
- ***ejecutarconsulta fichero***(String fichero) //miconsulta.xq(: Productos con precio > 50 y zona 10  
)

## XML:DB - Ejercicios

- Realizar aplicación que implemente el siguiente menú:

```
System.out.println(".....\n"
+ ". 1 Listar todos los departamentos. \n"
+ ". 2 Insertar un departamento (inserta 21, 'El 21',Vigo'. \n"
+ ". 3 Consultar un departamento.\n"
+ ". 4 Modificar un departamento (cambia el nombre por 'NOMBRE NUEVO').\n"
+ ". 5 Borrar un departamento.\n"
+ ". 0 SALIR.\n"
+ ".....\n");
```



## XML XQJ

- Para descargar la API accedemos a la URL: <http://xqj.net/exist/> y descargamos el fichero exist-***xqjapi-1.0.1.zip*** que contiene los ficheros *jar* que necesitamos importar a la aplicación Java.



exist-xqj-1.0.1.jar



exist-xqj-examples.jar



xqj2-0.0.1.jar



xqjapi.jar





## XML XQJ

- Interfaces

```
import javax.xml.xquery.XQConnection;  
import javax.xml.xquery.XQDataSource;  
import javax.xml.xquery.XQException;  
import javax.xml.xquery.XQPreparedExpression;  
import javax.xml.xquery.XQResultItem;  
import javax.xml.xquery.XQResultSequence;  
import net.xqj.exist.ExistXQDataSource;
```

## XML XQJ - Configurar una conexión

- ***XQDataSource***

```
XQDataSource server = new ExistXQDataSource();  
server.setProperty("serverName", "localhost");  
server.setProperty("port", "8080");  
server.setProperty("user", "admin");  
server.setProperty("password", "");
```

- ***XQConnection***

```
XQConnection conexion = server.getConnection();  
  
//También se puede indicar el usuario y password que abre la sesión:  
server.setProperty("password", "");  
  
//La conexión la cerramos con el método close():  
conexion.close();
```

## XML XQJ - Procesar resultados consulta

- *XQPreparedExpression*
- *XQResultSequence*

```
XQPreparedExpression consulta; //preparamos la consulta
XQResultSequence resultado;

consulta = conexion.prepareExpression("/EMPLEADOS/EMP_ROW[DEPT_NO = 10]");
resultado = consulta.executeQuery(); //ejecutamos la consulta
//recorremos el resultado para visualizarlo
while(resultado.next()){
    System.out.println("Elemento: " +resultado.getItemAsString(null));
}
```

## XML XQJ - Procesar resultados consulta

- ***XQPreparedExpression***
- ***XQResultSequence***

```
XQPreparedExpression consulta; //preparamos la consulta
```

```
XQResultSequence resultado;
```

```
consulta = conexion.prepareExpression("/EMPLEADOS/EMP_ROW[DEPT_NO = 10]");
```

```
resultado = consulta.executeQuery(); //ejecutamos la consulta
```

- ***XQResultItem***

```
XQResultItem resItem;
```

```
while(resultado.next()){// recuperamos los resultados con XQResultItem
```

```
    resItem = (XQResultItem) resultado.getItem();
```

```
    System.out.println("Elemento: " +resItem.getItemAsString(null)); }
```

## XML XQJ - Procesar resultados consulta

- *XQPreparedExpression*
- *XQResultSequence*

```
XQPreparedExpression consulta; //preparamos la consulta
```

```
XQResultSequence resultado;
```

```
consulta = conexion.prepareExpression("for $pro in /productos return $pro"); //consulta en xquery
```

- *XQSequence* - con 1 ó más *XQItem*

```
resultado = consulta.executeQuery(); //ejecutamos la consulta
```

```
while(resultado.next()){// recuperamos los resultados con XQResultItem
```

```
    System.out.println("Elemento: " +resultado.getItemAsString(null));
```

```
}
```



## XML XQJ - Ejercicios

- `verproductos();`//Ver to2 productos
- `cuentaproduc();`//"Numero de productos con precio > de 50: "
- `numporzona();`//"Numero de productos por cada zona "
- `ejecutarconsultadefichero();`//`miconsulta.xq`(: Productos con precio > 50 y zona 10 )  
(: Productos con precio > 50 y zona 10 :)  
*for \$prod in collection("/db/ColeccionesXML/BDProductosXML")/productos/produc*  
*where \$prod/precio>50 and \$prod/cod\_zona=10*  
*return \$prod*
- `creaemple10();`//`NUEVO_EMPLE10.xml`
- `muestraDatosProductosXML()` //Mostrar los datos del documento `productos.xml`, si tenemos productos en distintos documentos se mostrarán las etiquetas de todos los documentos



# Excepciones

- **API XML:DB**
  - XMLDBException
  - InstantiationException
  - NullPointerException
  - XMLDBException
    - *getMessage()*
- **API XQJ**
  - XQException
    - *getMessage()*