



# Linkia FP

Formación Profesional Oficial a Distancia



DAM – M06 – Clase 11

# 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:<br>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<br>C02        | Test UF1 | Examen UF1 |        |  |
|                                     | 14/02/2022 - 28/02/2022 | Tema 2. Persistencia en BDR con JDBC     | VC02                                      |                                    | C03<br>C04        | Test UF2 | Examen UF2 |        |  |
|                                     | 28/02/2022 - 14/03/2022 | Tema 3. Persistencia BDR con ORM         | VC03                                      | Actividad 2                        | C05<br>C06        |          |            |        |  |
|                                     | 14/03/2022 - 28/03/2022 | Tema 4. Persistencia en BDOO – BDOR      | VC04                                      | Actividad 3                        | C07<br>C08        |          |            |        |  |
| UF3: Persistencia en BD Nativas XML | 28/03/2022 - 19/04/2022 | Tema 5. Persistencia en BBDD nativas XML | VC05                                      | Actividad 4                        | C09<br>C10<br>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<br>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.

- Realización de consultas: clases y métodos.
- Tratamiento de excepciones.



# Realización de consultas

- XPath es el lenguaje de consultas sobre documentos XML (UF1).
- XQuery (soportado sobre XPath) es el lenguaje estándar propuesto por la W3C para ese fin.



CLASE

# XQuery (XML Query Language)

- XQuery es un lenguaje de consulta y procesamiento de datos XML propuesto por la W3C. Surge como equivalente natural de SQL pero para datos XML.
- XQuery es una extensión de Xpath 2.0.
- La definición completa de lenguaje XQuery es muy extensa y su especificación está formada por varios documentos.

CLASE

# XQuery (XML Query Language)

- FLWOR (For Let Where Order Return) es una sentencia que permite la unión de variables sobre conjuntos de nodos y la interacción sobre el resultado.

# XQuery (XML Query Language)

- FLWOR tiene la siguiente estructura:
  - La clausula FOR vincula una o más variables a expresiones escritas en XPath, creando un flujo de tuplas en el que cada tupla está vinculada a una de las variables.
  - La clausula LET vincula una variable al resultado completo de una expresión añadiendo esos vínculos de tuplas generadas por una clausula FOR o, si no existe ninguna clausula FOR, creando una única tupla que contenga esos vínculos.

CLASE

# XQuery (XML Query Language)

- La clausula WHERE permite establecer las condiciones que deben cumplir las tuplas devueltas como resultado.
- La clausula ORDER BY se utiliza para ordenar las tuplas devueltas según un criterio dado.
- La clausula RETURN muestra la estructura con la que se devolverá el resultado. Construye el resultado de la consulta para una tupla dada, después de haber sido filtrada por la clausula WHERE y ordenada por la clausula ORDER BY.

# XQuery (XML Query Language)

- Documento XML Base:

```
<!-- 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>
```

# XQuery (XML Query Language)

- FOR:
  - *for 'variable' in 'secuencia enlazada'*
  - *for \$a in //Libros/Libro/Autor  
return <MisAutores>{\$a}</MisAutores>*
- La consulta recupera todos los nodos *//Libros/Libro/Autor* que se identifican por la variable *\$a*.
- Cada una de las tuplas almacenadas en *\$a* se colocan entre las etiquetas *<MisAutores>*.

# XQuery (XML Query Language)

- FOR:

```
<MisAutores>
    <Autor>Nicolai Gogol</Autor>
</MisAutores>
<MisAutores>
    <Autor>Gonzalo Giner</Autor>
</MisAutores>
<MisAutores>
    <Autor>Umberto Eco</Autor>
</MisAutores>
```

# XQuery (XML Query Language)

- LET:
  - *let 'variable' := 'secuencia enlazada'*
  - *let \$a := //Libros/Libro/Autor*  
*return <MisAutores>{\$a}</MisAutores>*
- A diferencia de FOR, LET obtiene un única tupla y no varias.
- El resultado serán todos los autores en una única etiqueta *<MisAutores>*.

CLASE

# XQuery (XML Query Language)

- LET:

```
<MisAutores>
    <Autor>Nicolai Gogol</Autor>
    <Autor>Gonzalo Giner</Autor>
    <Autor>Umberto Eco</Autor>
</MisAutores>
```

# XQuery (XML Query Language)

- WHERE:
  - *WHERE 'expresion'*
  - *FOR \$a in //Libros/Libro/Autor*  
*WHERE upper-case(substring(\$a, 1, 7))='NIKOLAI'*  
*return <LibrosNikolai>{\$a/ancestor::Libro/Titulo}*  
*</LibrosNikolai>*

# XQuery (XML Query Language)

- ‘expresión’ se evalua a booleano para cada tupla del flujo:
  - Si el resultado es true, se trata.
  - Si el resultado es false, se descarta.
- La forma de obtener un booleano de la expresión es igual que en la sentencia condicional de XPath 2.0.

CLASE

# XQuery (XML Query Language)

- El ejemplo recupera todos los <Titulo> de los libros cuyo <Autor> tenga entre sus primeros 7 caracteres la cadena ‘NIKOLAI’ (en mayúsculas).
- La consulta utiliza *ancestor::* para hacer referencia al padre de \$a y así poder acceder al Titulo del Libro.

```
<LibrosNikolai>
    <Titulo>El Capote</Titulo>
</LibrosNikolai >
```

# XQuery (XML Query Language)

- ORDER BY:
  - *for \$a at \$p in //Libros/Libro  
where \$p>1 and \$p<4  
order by \$a descending  
return <MisLibros posicion="{\\$p}">  
{\$a/Autor} </MisLibros>*
- La consulta obtiene los autores de los libros cuya posición es mayor que 1 y menor que 4.
- El resultado es ordenado por la posición descendiente.

CLASE

# XQuery (XML Query Language)

- ORDER BY:

```
<MisLibros posicion="3">
    <Autor>Umberto Eco</Autor>
</ MisLibros >
<MisLibros posicion="2">
    <Autor>Gonzalo Giner</Autor>
</ MisLibros >
```

# XQuery (XML Query Language)

- RETURN:
  - *for \$a at \$p in //Libros/Libro  
where \$p=1  
order by \$a descending  
return element MisLibros {attribute posicion {\$p},  
attribute titulo {\$a/Titulo}}*
- La potencia de esta sentencia radica en que permite la creación de nuevos elementos.  
  
*<MisLibros posicion="1" titilo="El Capote"/>*

# Ejecutar consultas XQuery con XML:DB

- Las consultas *XQuery* se ejecutan con XML:DB en eXist de la misma manera que lo hacen las consultas XQuery Update Extension.
- Las clases que intervienen a la hora de hacer una consulta son: *Collection*, *XQueryService*, *ResourceSet* y *ResourceIterator*.

CLASE

# Ejecutar consultas XQuery con XQJ

- Para realizar consultar con XQJ es necesario hacer previamente una conexión a la BBDD.
- Las siguientes clases y métodos son los que utilizamos para ejecutar una consulta XQuery:
  - *XQExpression* contiene la funcionalidad necesaria para definir expresiones que se ejecutarán generalmente con *executeQuery* o *executeCommand*.

CLASE

# Ejecutar consultas XQuery con XQJ

- *XQResultSequence* es una clase de tipo *XQSequence* que contiene una secuencia de ítems que pueden ser recorridos.

# Tratamiento de excepciones

- Vamos a detallar las clases que definen las excepciones y los mensajes de error devueltos por cada sistema gestor.

# Excepciones en XML:DB

- La clase *XMLDBException* captura todos los errores que se producen al tratar con BBDD mediante XML:DB.
- La clase *ErrorCodes* define los códigos de error de XML:DB usados por el atributo *errorCodes* de una excepción *XMLDBException*.

CLASE

# Errores XMLDBException

- **COLLECTION\_CLOSED**

Se activa si se ha llamado al método close en Collection.

- **INVALID\_DATABASE, INVALID\_COLLECTION,  
INVALID\_RESOURCE, INVALID\_URI**

Se activan para indicar que la BBDD, colección, recurso o URI son inválidos.

CLASE

# Errores XMLDBException

- NO SUCH DATABASE, NO SUCH COLLECTION, NO SUCH RESOURCE y NO SUCH SERVICE  
Se activan si la BBDD, la colección, el recurso o la URI no pueden ser localizados.
- Otros errores:  
NOT\_IMPLEMENTED, VENDOR\_ERROR, UNKNOWN\_ERROR, UNKNOWN\_RESOURCE\_TYPE, VENDOR\_ERROR, PERMISSION\_DENIED, WRONG\_CONTENT\_TYPE.

CLASE

# Excepciones en XQJ

- La clase XQException captura todos los errores que se producen al tratar con BBDD mediante XQJ.
- Dentro de XQJ hay una cadena que informa del error que se ha capturado.
- Este error es tratado como un mensaje Java Exception, disponible en *getMessage()*.

## Excepciones en XQJ

- La causa del error está disponible con el método *getCause()*.
- Los errores propios del sistema gestor (fabricante-vendor) son obtenidos con *getVendorCode()*.
- Todos los errores detectados están disponibles en forma de cadena de objetos XQException.

CLASE

## Actividad 04

Base de datos con XND.

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

CLASE

# Final UF3

- Actividad
  - Fecha de entrega: 19/04/2022
- Preguntas evaluación de clase
- Test



# Linkia FP

Formación Profesional Oficial a Distancia