

# **Introducción a la base de datos XML: eXistDB**

---

**Calbet González, María Victoria**

# Índice

---

1. Introducción
2. Objetivos
3. Tipos de base de datos XML
4. ¿Qué es eXistdb?
5. XQuery
6. Herramientas

# 1. Introducción

---



## Objetivos

- Formal y conciso
- Extensible
- Fácil de leer y editar
- Fácil de implantar, programar y aplicar a los distintos sistemas

## 2. Objetivos

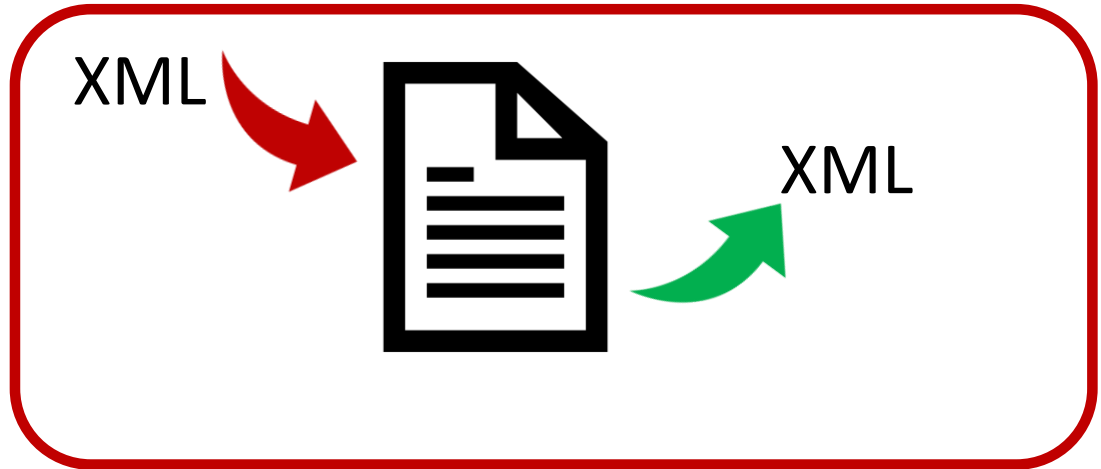
---

- Manejar el uso de la base de datos eXistdb para XML.
- Aprender y manejar de forma básica Xquery.
- Aplicar este tipo de bases de datos al uso cotidiano en herramientas para la ingeniería de software.
- Distinción entre las posibilidades que ofrecen unas bases de datos y otras, para saber cuál se adapta mejor a nuestras necesidades.

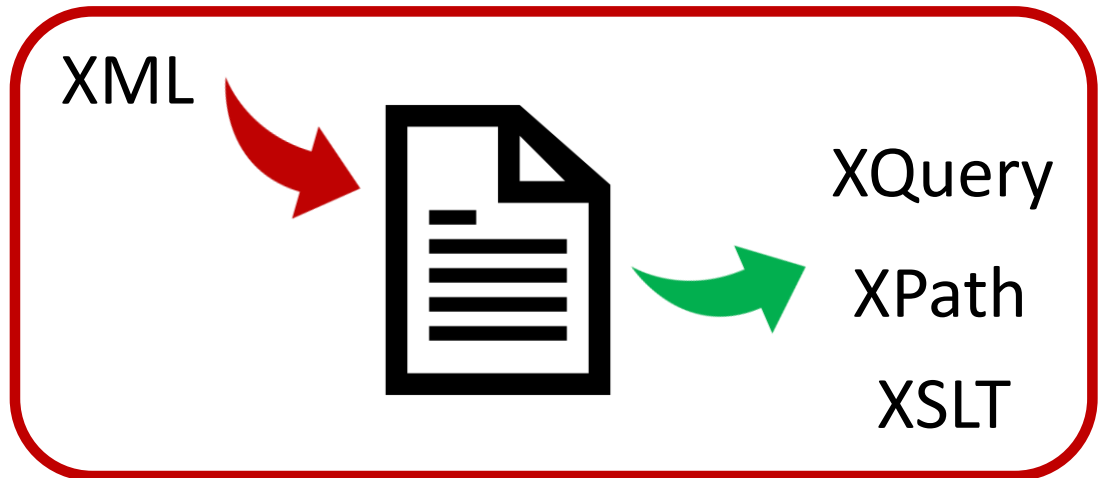
### 3. Tipos de base de datos XML

---

#### XML HABILITADO



#### XML NATIVO



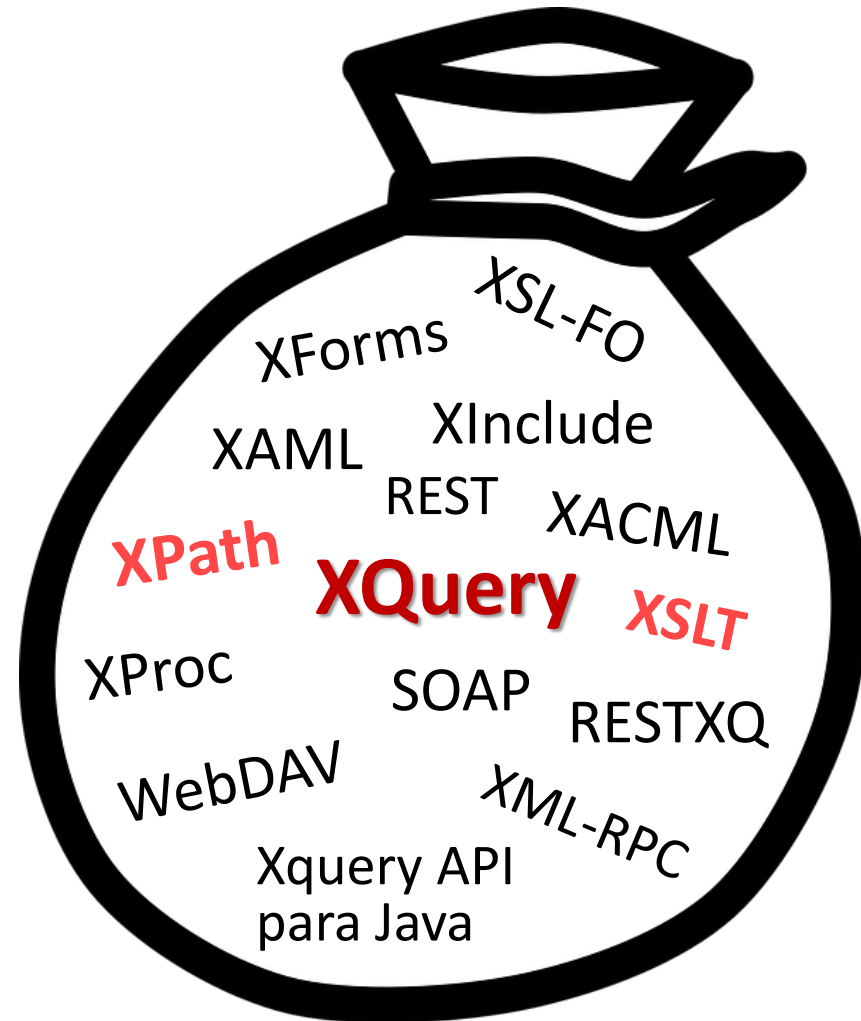
## 4. ¿Qué es eXistdb?

---



Código abierto

Base de datos XML nativa





Backup



betterFORM Demo  
and Reference XForms



Collections



eXide - XQuery IDE



eXist-db Demo Apps



eXist-db  
Documentation



eXist-db Public  
Application Repository



Java Admin Client



Markdown Parser in  
XQuery



Monitoring and Profiling  
for eXist (Monex)





File Edit Navigate Buffers Application XQuery Help Login

New New XQuery Open Save Close Eval Run

Current app: unknown

File Type: XQuery

outline directory

new-document 2\*

```
1 xquery version "3.1";
2
3 (: Declaración de los namespaces :)
4 declare namespace rem = "http://rem.lsi.us.es";
5
6 (: Visualizar el xml completo que vamos a usar, el documento :)
7 (:let $r := doc("/db/apps/CBD/REM/Requisitos.xml")/*;)
8 (: return $r;)
9
10
11 (: Devuelve los nombres de los stakeholders :)
12 let $t := doc("/db/apps/CBD/REM/Requisitos.xml") (: se guarda en la variable el documento :)
13 let $p := $t//element(rem:stakeholder)//element(rem:name) (: Se guarda en la variable el nombre de los stakeholders :)
14 let $r := distinct-values($p) (: Si hay algun elemento repetido, lo devuelve una sola vez :)
15 for $ste in $r (: recorre los nombres :)
16 order by $ste (: Ordena por nombre :)
17 | return <name>{$ste}</name>
18
19 (:(: Dar los nombres de los objetivos :):)
20 (:for $t in doc("/db/apps/CBD/REM/Requisitos.xml") :)
21 (:let $e := $t//element(rem:section)[@oid="SEC-0025"]//element(rem:objective)//element(rem:name) (: Recorre el documento y :)
22 (:busca la seccion que tenga un atributo SEC-0025 dando el nombre del objetivo :):)
23 (:return:)
24 (: <objetivos> {$e} </objetivos>:)
25
26 (:(: Devuelve aquellos stakeholders que el día de su creacion sea menor o igual a 5 y lo concatena con el nombre q le pertenece y la
fecha :):)
27 (: :)
28 (: for $b in doc("/db/apps/CBD/REM/Requisitos.xml")//element(rem:stakeholder) (: recorre todos los stakeholders :):)
29 (: where $b//element(rem:day)<=5 (: dia menor o igual a 5 :):)
30 (: return <resultado> {concat($b//element(rem:date),"/", $b//element(rem:name))}</resultado> (: da el resultado :)

```

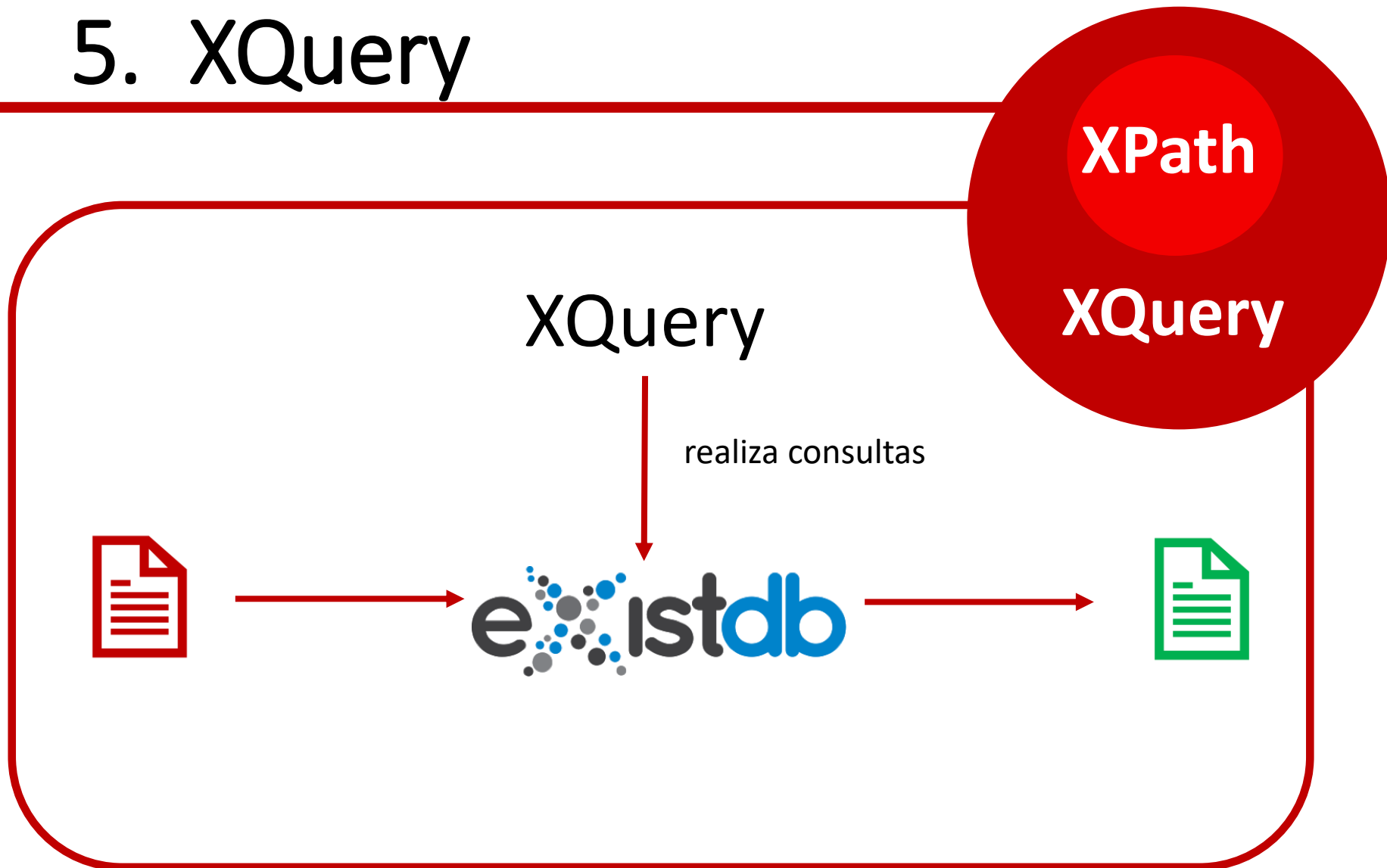
\_new\_2

XQuery returned 5 item(s) in 1 nombre  
0.024s

Filter by...



## 5. XQuery



**For** → Crea un flujo de tuplas

**Let** → Vincula una variable con respecto a la tupla generada por ***for***

**Where** → Filtra

**Order by** → Ordena

**Return** → Construye

## 6. Herramientas

---



Agencia Estatal de Meteorología



Agencia Tributaria

# BPMN

```
1 <bpmn:definitions xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL" xmlns:di="http://www.omg.org/spec/BPMN/20100524/DI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" targetNamespace="http://bpmn.io/schema/bpmn">
2   <bpmn:collaboration id="Collaboration_1p17tfo">
3     <bpmn:participant id="Participant_18ndlvj" name="Integridad de las transmisiones de una e
4     ="Process_1j3ipq7"/>
5   </bpmn:collaboration>
6   <bpmn:process id="Process_1j3ipq7">
7     <bpmn:laneSet>
8       <bpmn:lane id="Lane_1821liv" name="Portal bancario&#xA;">
9         <bpmn:flowNodeRef>Task_0nrqdbd</bpmn:flowNodeRef>
10        <bpmn:flowNodeRef>ExclusiveGateway_0a6m49v</bpmn:flowNodeRef>
11        <bpmn:flowNodeRef>Task_0m8ku0u</bpmn:flowNodeRef>
12        <bpmn:flowNodeRef>Task_0th3a9v</bpmn:flowNodeRef>
13        <bpmn:flowNodeRef>Task_1ilhvfs</bpmn:flowNodeRef>
14        <bpmn:childLaneSet xsi:type="bpmn:tLaneSet">
15          <bpmn:lane id="Lane_0dizr4j8" name="&#xA;">
16            <bpmn:flowNodeRef>Task_0nrqdbd</bpmn:flowNodeRef>
17            <bpmn:flowNodeRef>ExclusiveGateway_0a6m49v</bpmn:flowNodeRef>
18            <bpmn:flowNodeRef>Task_0m8ku0u</bpmn:flowNodeRef>
19            <bpmn:flowNodeRef>Task_0th3a9v</bpmn:flowNodeRef>
20            <bpmn:flowNodeRef>Task_1ilhvfs</bpmn:flowNodeRef>
21          </bpmn:lane>
22        </bpmn:childLaneSet>
23      </bpmn:lane>
24      <bpmn:lane id="Lane_1hm7zkn" name="Cliente bancario&#xA;">
25        <bpmn:flowNodeRef>Task_1oq97cn</bpmn:flowNodeRef>
26        <bpmn:flowNodeRef>Task_1549xu7</bpmn:flowNodeRef>
27        <bpmn:flowNodeRef>StartEvent_1ph5197</bpmn:flowNodeRef>
28        <bpmn:flowNodeRef>Task_0uvio8o</bpmn:flowNodeRef>
29        <bpmn:flowNodeRef>Task_0y1v89r</bpmn:flowNodeRef>
```

# REM

```
1 <rem:requirementsProject xmlns:rem="http://rem.lsi.us.es">
2   <rem:name>Requisitos</rem:name>
3   <rem:c_requirementsSpecification oid="CRS-0001">
4     <rem:name>Documento de Requisitos del Sistema</rem:name>
5     <rem:version>
6       <rem:major>1</rem:major>
7       <rem:minor>0</rem:minor>
8     <rem:date>
9       <rem:year>2015</rem:year>
10      <rem:month>10</rem:month>
11      <rem:day>19</rem:day>
12    </rem:date>
13  </rem:version>
14  <rem:comments>
15    Ninguno
16  </rem:comments>
17    <rem:preparedFor organizations="ORG-0001 "/>
18    <rem:preparedBy organizations="ORG-0002 "/>
19  <rem:section oid="SEC-0001">
20    <rem:name>
21      Introducción</rem:name>
22    <rem:version>
23      <rem:major>1</rem:major>
24      <rem:minor>0</rem:minor>
25    <rem:date>
26      <rem:year>2015</rem:year>
27      <rem:month>10</rem:month>
28      <rem:day>19</rem:day>
29    </rem:date>
30    </rem:version>
31  <rem:comments>
```

# REM

```
(: Dar los nombres de los objetivos :)
for $t in doc("/db/apps/CBD/REM/Requisitos.xml")
let $e := $t//element(rem:section)[@oid="SEC-0025"]//element(rem:objective)//element(rem:name) (: Recorre el documento y
busca la seccion que tenga un atributo SEC-0025 dando el nombre del objetivo :)
return
  <objetivos> {$e} </objetivos>
```

```
1 <objetivos>
  <rem:name xmlns:rem="http://rem.lsi.us.es"> Gestión de stock</rem:name>
  <rem:name xmlns:rem="http://rem.lsi.us.es"> Gestión de usuarios</rem:name>
  <rem:name xmlns:rem="http://rem.lsi.us.es"> Gestión de pedidos</rem:name>
</objetivos>
```

# BPMN

```
(: Devuelve aquellas tareas una x e y especifica :)
for $b in doc("/db/apps/CBD/BPMN/bpmn.xml")//element(bpmndi:BPMNShape)
let $f := doc("/db/apps/CBD/BPMN/bpmn.xml")//element(bpmn:task)
where some $a in $b satisfies ((($a//element(dc:Bounds)/@x < 410) and ($a//element(dc:Bounds)/@y > 100))
rango de x e y:)
let $e := data($b/@bpmnElement) (: devuelve el atributo bpmnElement:)
where starts-with($e,"Task_") (: si en $e algun elemento guardado que empiece con "task_":)
return $e (: devuelve aquellos que dependen de la comparacion:)
```

1 Task\_1549xu7

2 Task\_1oq97cn

3 Task\_0nrqdbd

*¡Gracias!*

