Práctica XQuery

BaseX & eXist

REPOSITORIOS DE INFORMACION

Grado en Ingeniería Informática del Software

Escuela de Ingeniería Informática
Universidad de Oviedo

José Antonio Sánchez Sánchez
sanchezsjose@uniovi.es

Trabajo a entregar

- Grupos de 3 Personas y de Temática Libre
- Obligatorio creación de al menos 9 consultas XQuery:
  - 3 consultas simples con condiciones
  - 3 consultas FLWOR
  - 3 consultas FLWOR anidadas o múltiples
- Se valorará:
  - Adecuación de la temática
  - Complejidad y diversidad de las consultas
- Informe que indique:
  - Descripción de los ficheros XML seleccionados
  - Porcentaje de participación y tareas realizadas por cada miembro
  - Objetivo de cada consulta
  - Solución propuesta
  - Resultado de cada consulta (resumido si es muy largo)
- Fecha Límite de entrega
  - 12/11/2017

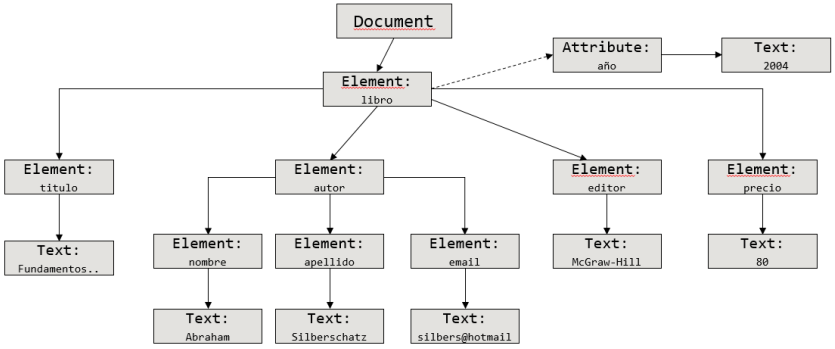
Antes de Empezar…

- Temática Libre → Alumno definirá el objetivo del proyecto y buscará los ficheros XML adecuados par alcanzar el objetivo
- Obligatorio creación de al menos 9 consultas XQuery:
  - 3 consultas simples con condiciones
  - 3 consultas FLWOR
  - 3 consultas FLWOR anidadas o múltiples
- Ejemplos de Fuentes de Archivos XML:
  - Datos del Gobierno de España: http://datos.gob.es/catalogo
  - Datos del Ayto. de Gijón: https://transparencia.gijon.es/page/1808-catalogo-de-datos
  - Otros de tipo “Open Access”
- Ver Fichero “Entregable XQuery” del Campus Virtual

¿Alguna Duda?

Repaso Rápido

- XQuery → Lenguaje de consulta sobre ficheros XML basado en un modelo de datos en forma de árbol
- Basado en Descriptores de Camino (XPath) / y //
- Símbolo @ para atributos
- Predicados [ ] → Condiciones
  - [titulo = “Fundamentos”]
- doc(“bib.xml”)/bib/libro[1]/autor



- XQuery es sensible a mayúsculas → comentarios (: :)
- Consultas entre llaves {} para evitar literal en consulta
- Consulta simple con condición de apellido:

```
doc("bib.xml")/bib/libro[autor/apellido="Korth"]
```

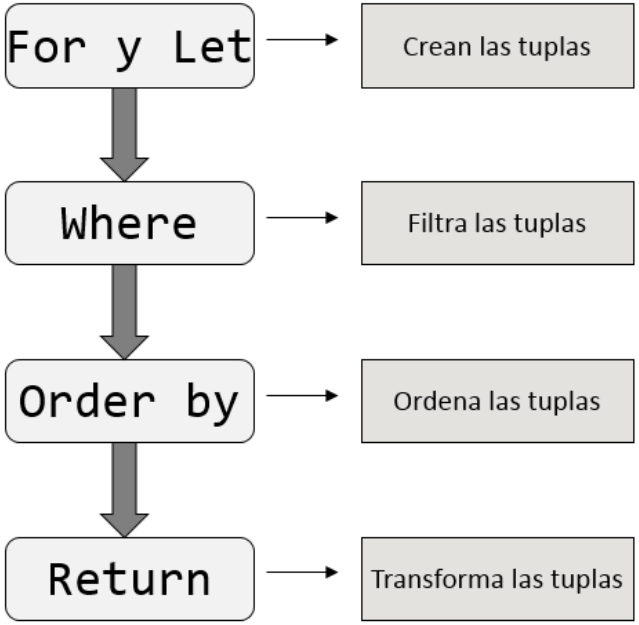
Repaso Rápido (II)

- Formato Final de Consultas para práctica

```
<ejemplo>
    <p> Esto es una consulta de ejemplo </p>
    <ej> doc("bib.xml")//libro[1]/titulo </ej>
    <p> Este es el resultado de la consulta anterior </p>
    <ej> Fundamentos de Bases de Datos </ej>
</ejemplo>
```

- Expresiones FLWOR (***F**or **L**et **W**here **O**rder **R**eturn*)

```
for $b in doc("bib.xml")//libro
  where $b/@año="2000"
  return $b/titulo
```



BaseX

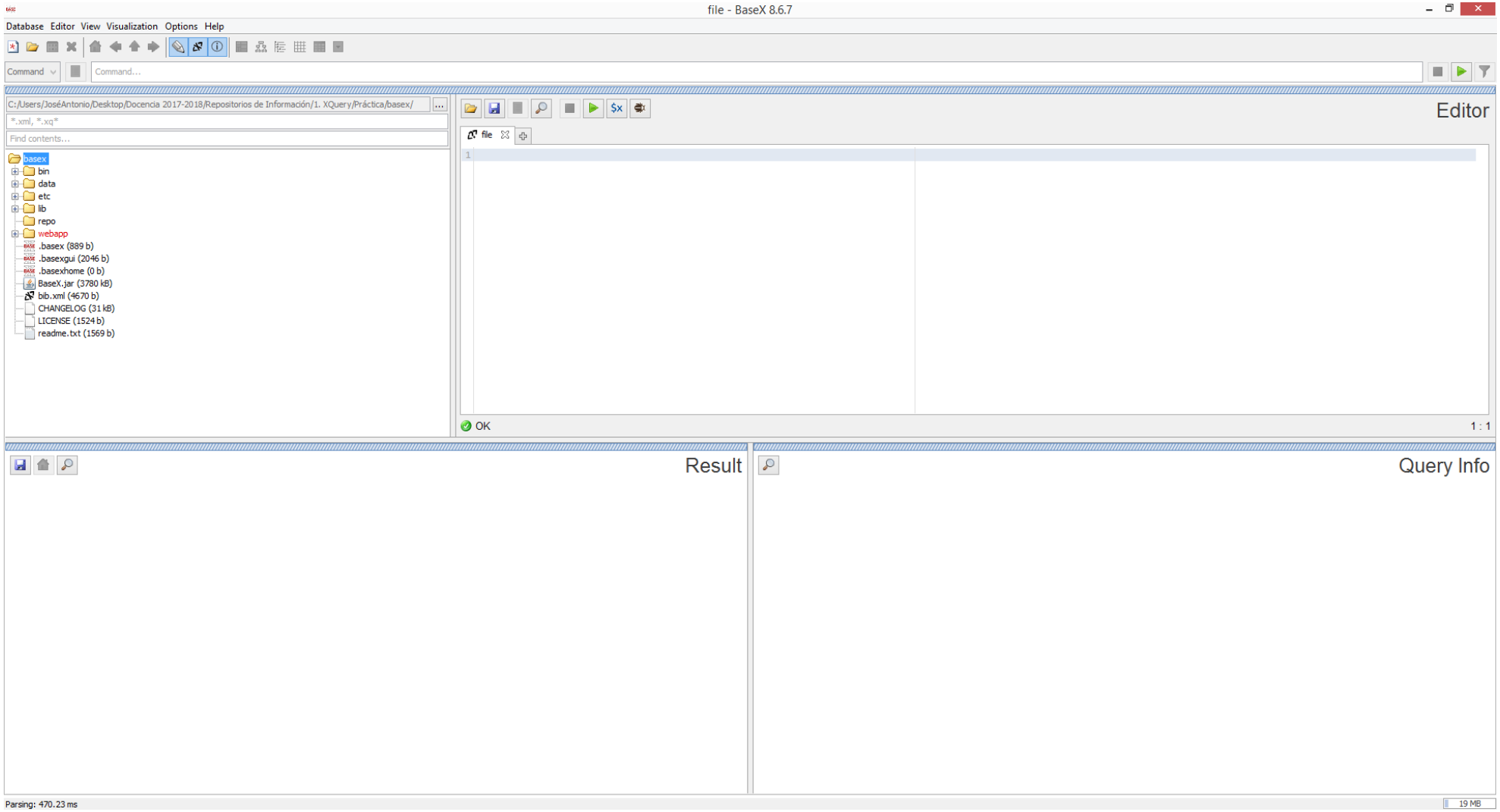
- Motor de Base de Datos XML

http://basex.org/

BaseX - Propiedades

- Arquitectura Cliente/Servidor
- Escalable y de alto rendimiento
- Muy ligero
- Con procesamiento de XPath/XQuery
- Soporta últimas actualizaciones de W3C
- Maneja operaciones simultáneas de lectura y escritura con multiples usuarios
- Interfaz visual para explorar datos y resultados

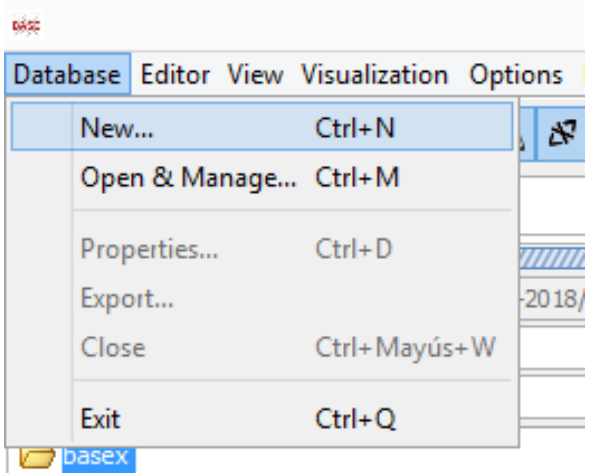
BaseX - Interfaz

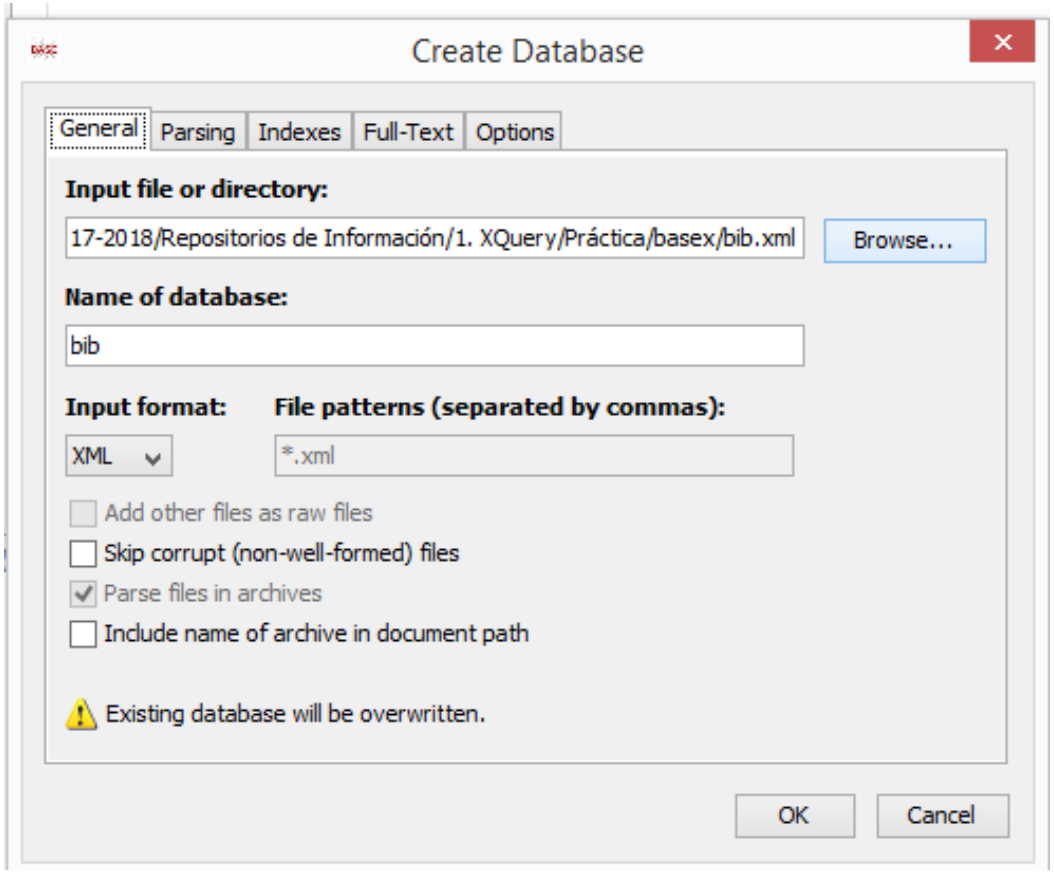


Empezando a Trabajar con BaseX

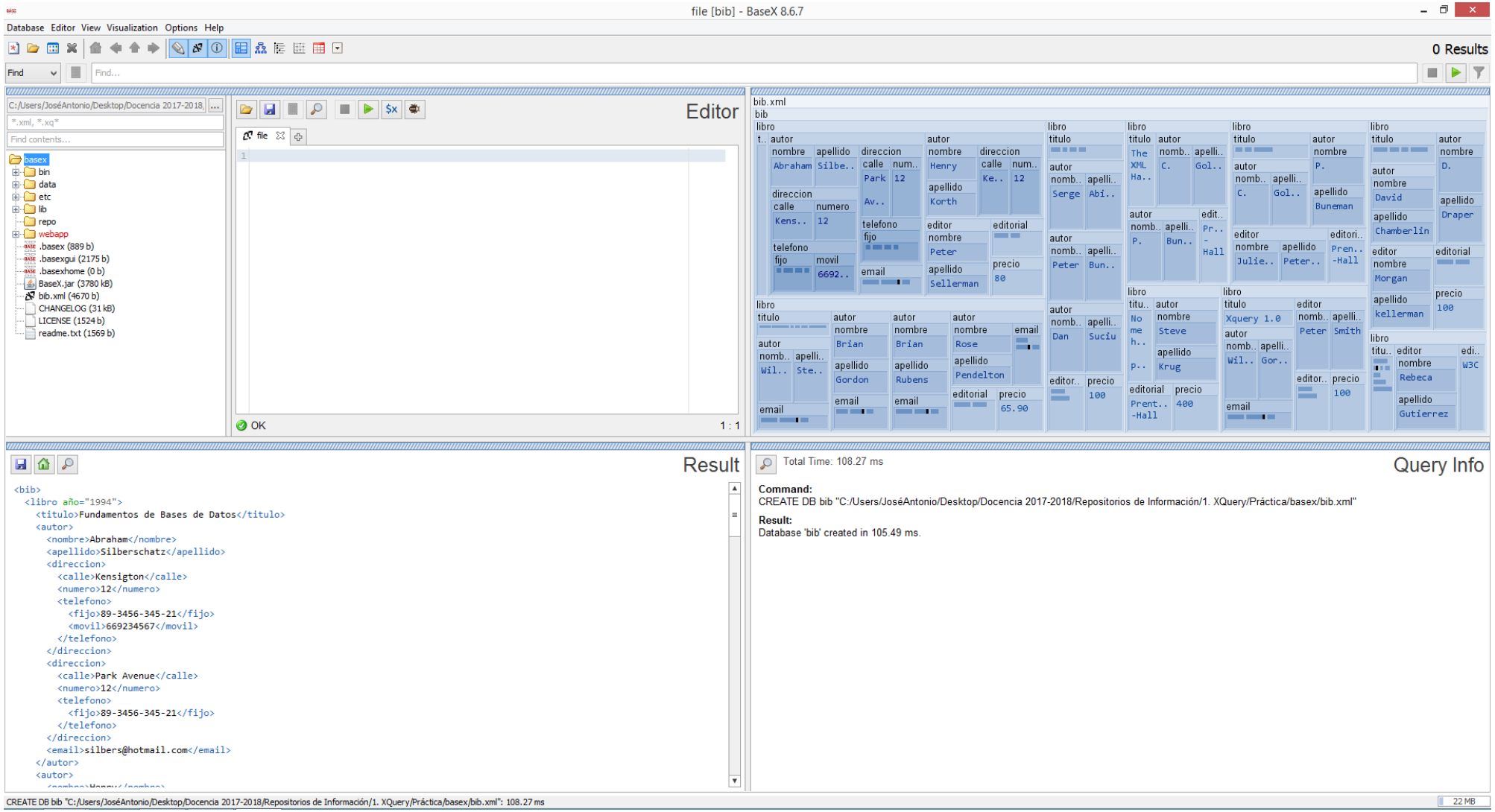
- Base de datos XML
- Necesidad de fichero XML asociado con contenido (*.xml)
- Ejecutar BaseX (*BaseX.jar*) para empezar a trabajar con el entorno

Carga de Fichero XML



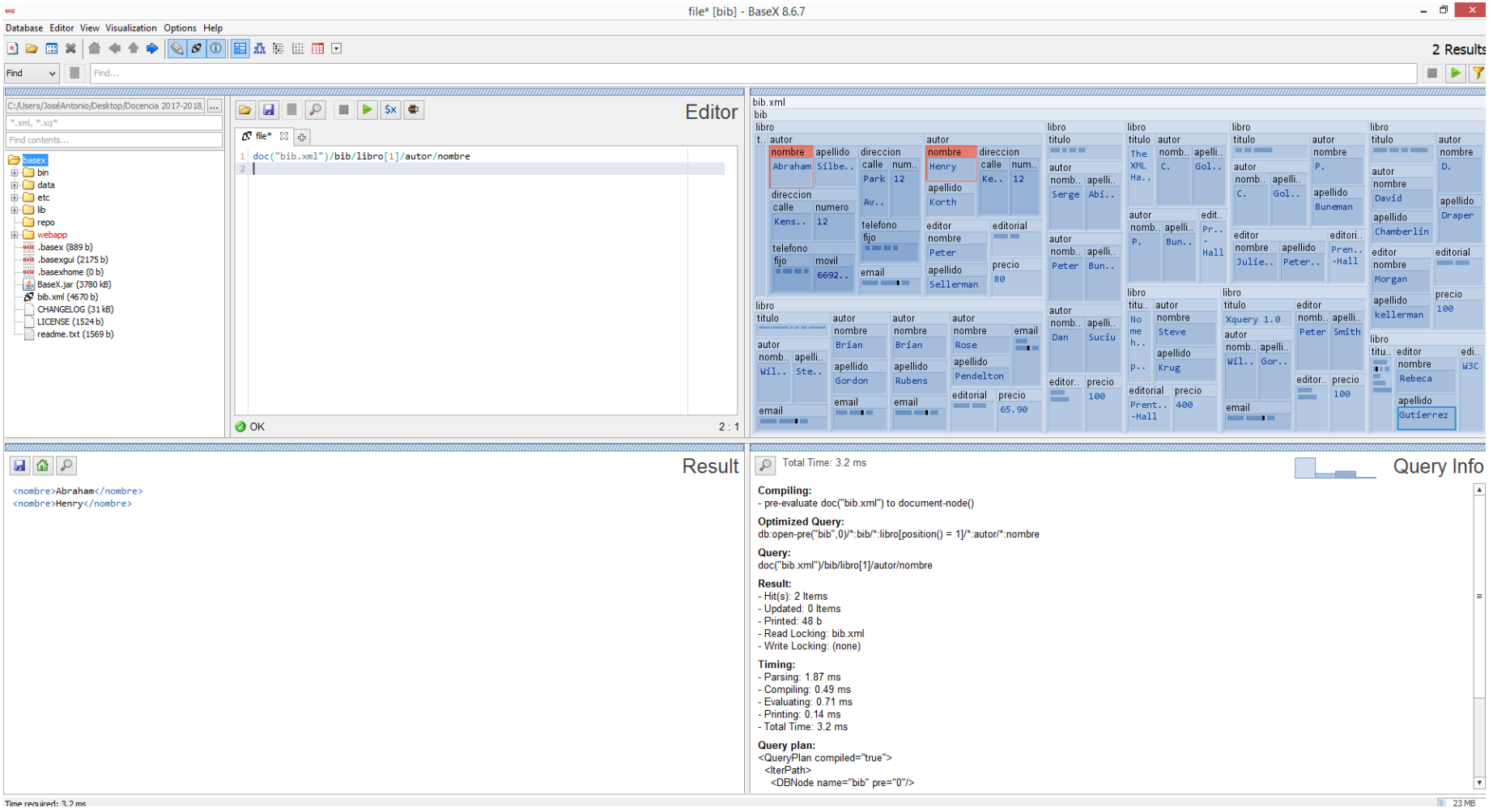


Fichero XML cargado



Lanzando consultas

- Una vez cargado el fichero XML escribimos la consulta correspondiente para ver el resultado y la información



Ejemplos de Consultas Sencillas

- Utilizamos el fichero de ejemplo “bib.xml” para lanzar las consultas

➢ Selecciona la lista de todos los títulos de los libros que tengan más de 2 autores

```
doc("bib.xml")/bib/libro[count(autor)>2]/titulo
```

➢ Devuelve el precio, incrementado en un 50%, de los libros que contengan la palabra “Xquery”

```
/bib/libro[contains(titulo,"Xquery")]/(precio*1.21)
```

➢ Devuelve la cantidad total de autores que existen

```
count(doc("bib.xml")//autor)
```

Ejemplos de Consultas FLWOR

- De nuevo utilizamos el fichero bib (bib.xml)

➢ Devuelve la lista de editores de toda la colección ordenados ascendentemente

```
for $a in doc("bib.xml")//editor
  order by ($a) ascending
    return $a
```

➢ Lista el título de cada libro junto con el número de autores y editores que tiene

```
for $b in doc("bib.xml")//libro
  let $c := $b/autor
  let $d := $b/editor
  return <libro> { $b/titulo, <numero_autor> {count ($c)} </numero_autor>,
    <numero_editor> {count ($d)} </numero_editor> }</libro>
```

Ejemplos de Consultas FLWOR (múltiples ficheros)

- Utilizamos el fichero bib.xml y amazon.xml
- Se deben cargar las bases de datos xml en BaseX para poder utilizarlas

➢ Títulos de libros y precio para aquellos que tengan unos precios inferiores o iguales en amazon que en bib

```
for $i in doc("bib.xml")//libro
  for $j in doc("amazon.xml")//libro
    where $i/titulo = $j/titulo and $i/precio>=$j/precio
       return
        <libros-precios-baratos-amazon>
         {
           $i/titulo,
           <precio-amazon>{ string($j/precio) }</precio-amazon>,
           <precio-bib> { string($i/precio) }</precio-bib>
         }
        </libros-precios-baratos-amazon>
```

Ejemplos de Consultas con Salida HTML

- Devuelve los títulos de los libros en forma de lista

```
<html>
  <ul>
    {
       for $x in doc("bib.xml")//titulo
       order by $x
       return <li>{$x}</li>
     }
</ul>
</html>
```

- Crea un documento con extensión .html y copia y pega el código resultante de ejecutar la consulta superior. ¿Qué ocurre?

Ejemplos de Consultas con Salida HTML (II)

- Listar los títulos de los libros del documento bib.xml en formato de tabla con borde

```
<html>
   <head> <title> Titulos de libros seleccionados </title>
  </head>
  <body>
       <table border="1">{
             for $b in doc("bib.xml")//libro
              return
                   <tr> <td> <i> {string($b/titulo)} </i> </td> </tr>}
       </table>
  </body>
</html>
```

- De nuevo, vuelve a crear un fichero con extensión .html y copia y pega el código resultante del lanzamiento de la consulta. ¿Qué ocurre?

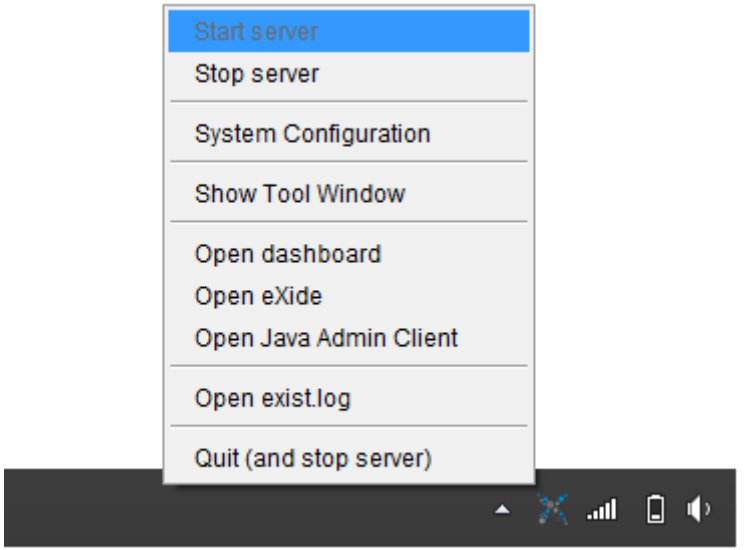
eXist

- Base de datos open-source nativa XML creada por Wolfgang Meier (2000)

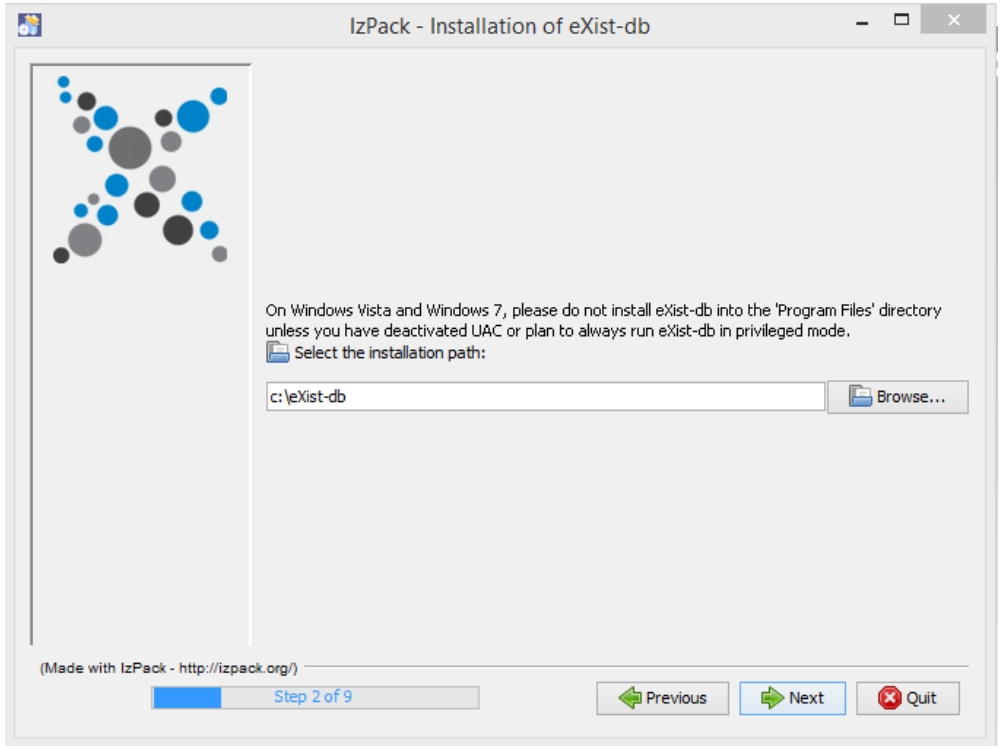
- Características Fundamentales:
  - Almacenamiento basado en **árboles B+** y ficheros paginados. Los nodos documento son almacenados en DOM persistentes.
  - Los documentos se organizan en **colecciones jerárquicas**. Las colecciones no están ligadas a esquemas predefinidos o tipos documento.
  - Proporciona una **indexación por defecto**
  - Proporciona un motor propio y optimizado para **XQuery**
  - Proporciona actualizaciones a nivel de documento y de nodo mediante **XUpdate y XQuery**
  - Puede ser desplegada como un servidor de base de datos stand-alone, como una librería Java embebida o como parte de una aplicación web (ejecutándose en el motor de servlets).
  - Proporciona utilidades **backup/restore**
  - http://exist-db.org

Instalación

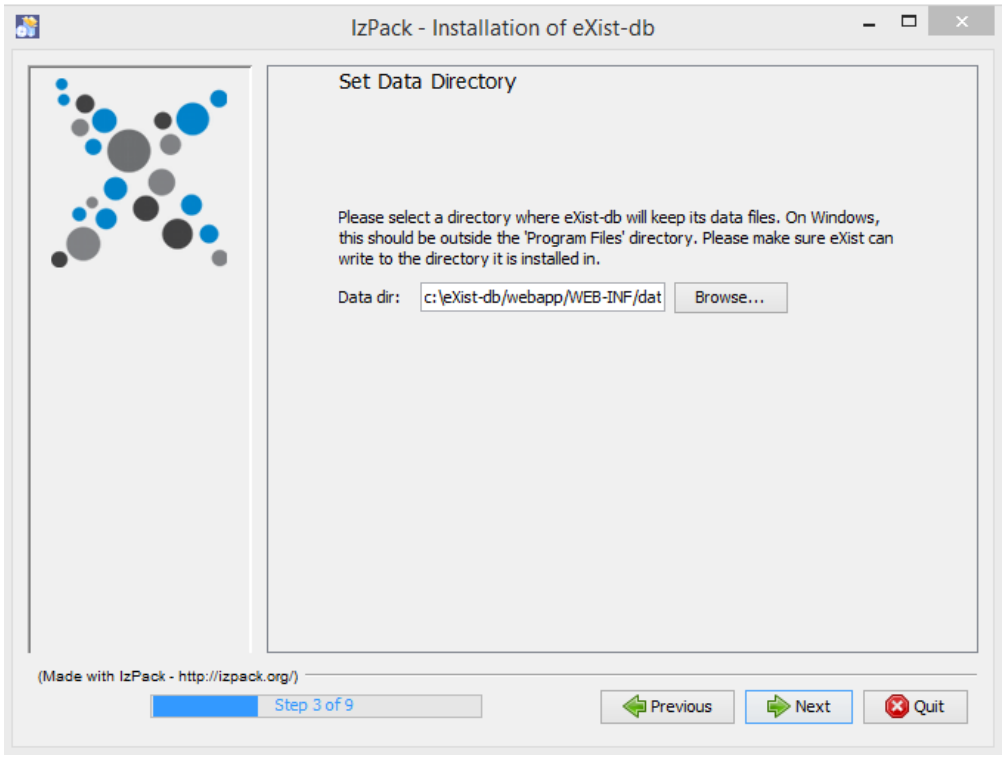
- Última versión 3.5.0
  - https://bintray.com/existdb/releases/exist/3.5.0/view
  - `java -jar eXist-db-setup-3.5.0.jar`
- Arrancar la base de datos
  - Acceso directo eXist-db Database
  - `java -jar start.jar`
- Administración del servidor
  - Menú contextual try icon
  - Cliente Java
    - Administración de usuarios
- Acceder al servidor: *dashboard*
  - http://localhost:8080/

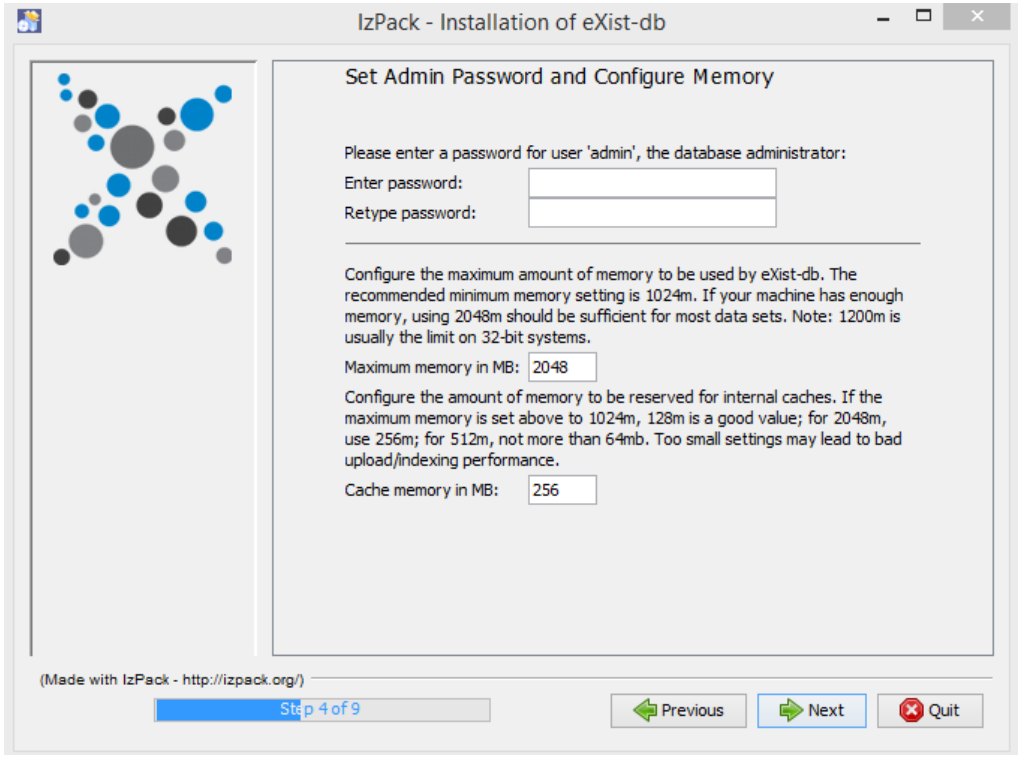


Instalación (II)

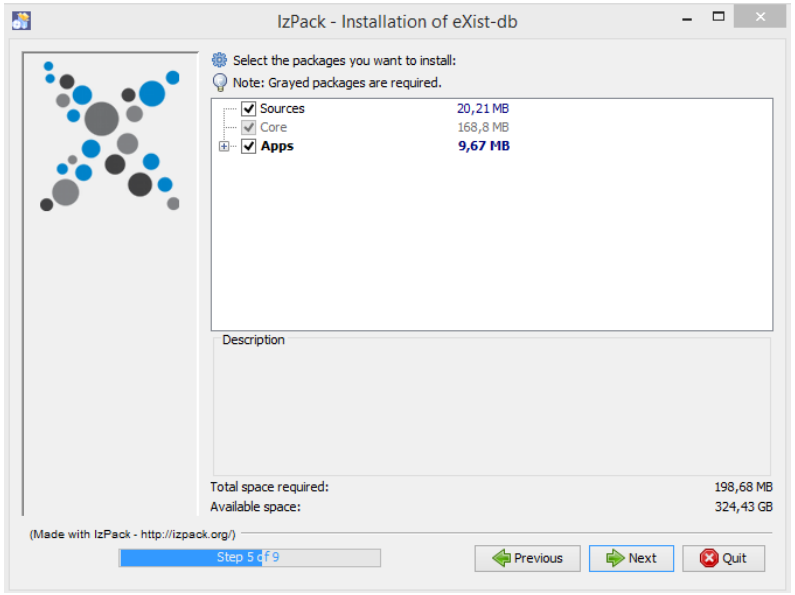


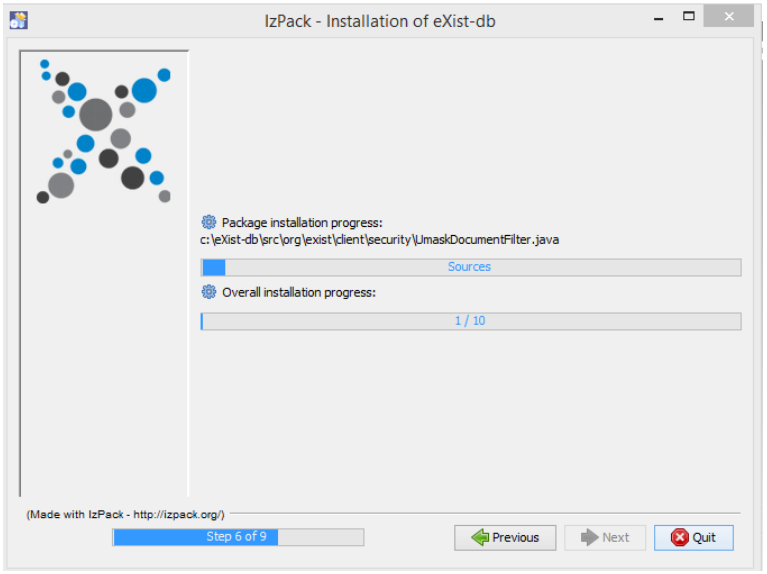
Instalación (III)

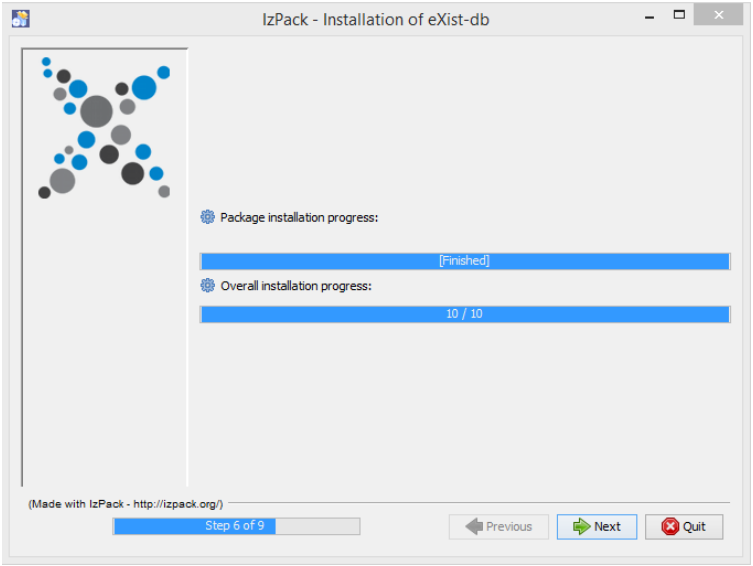




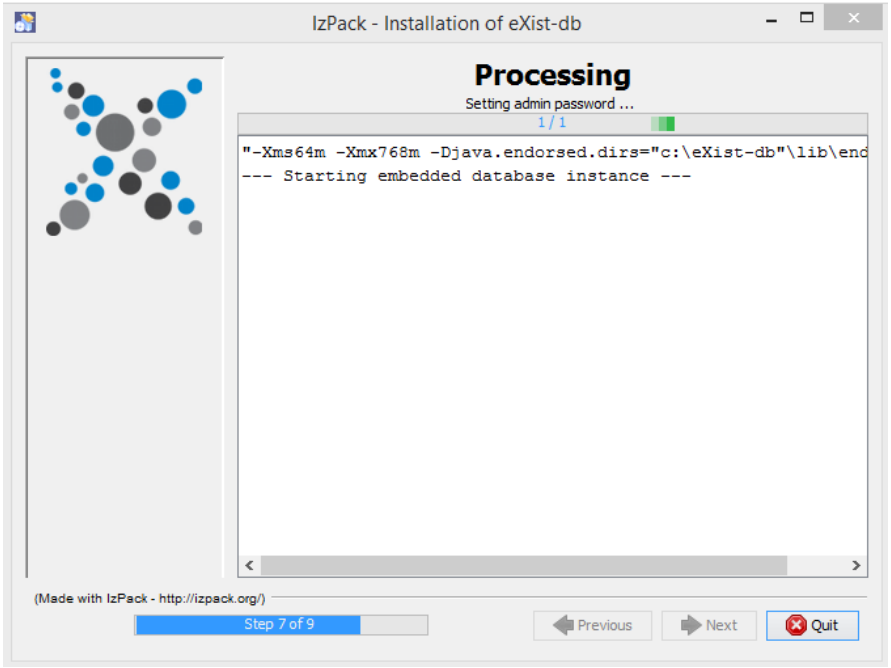
Instalación (IV)

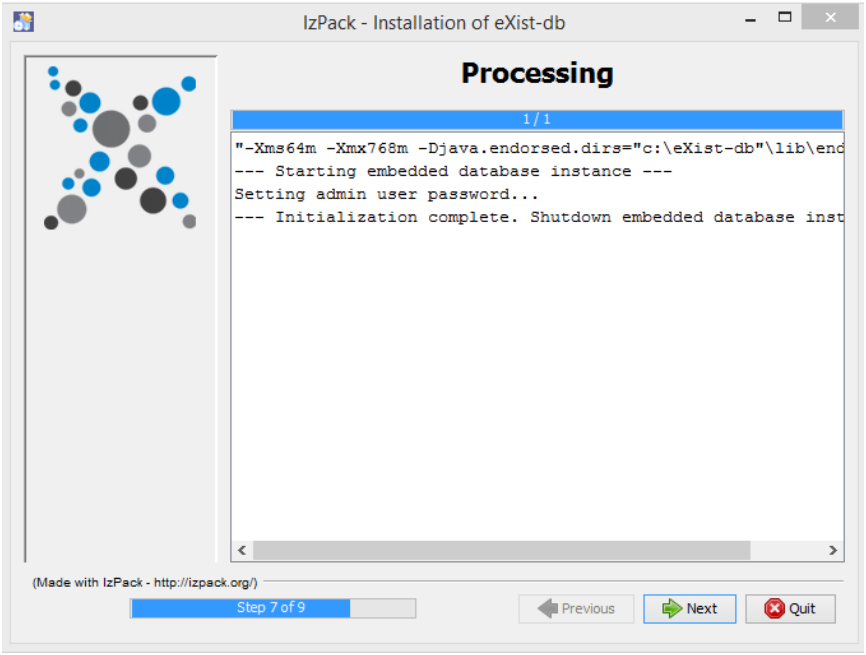




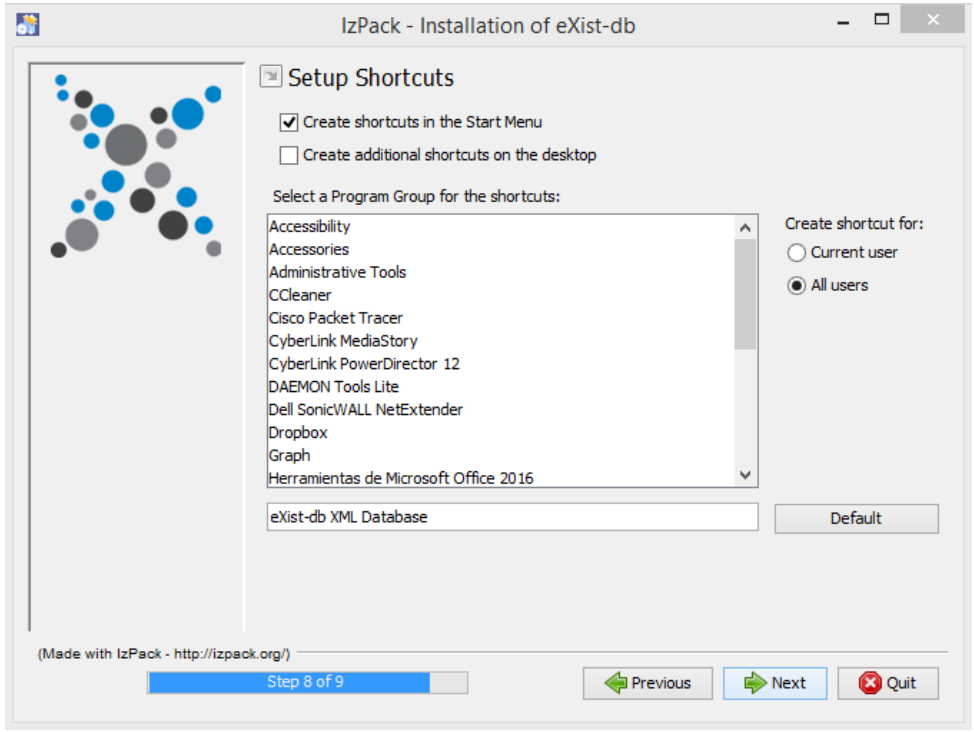


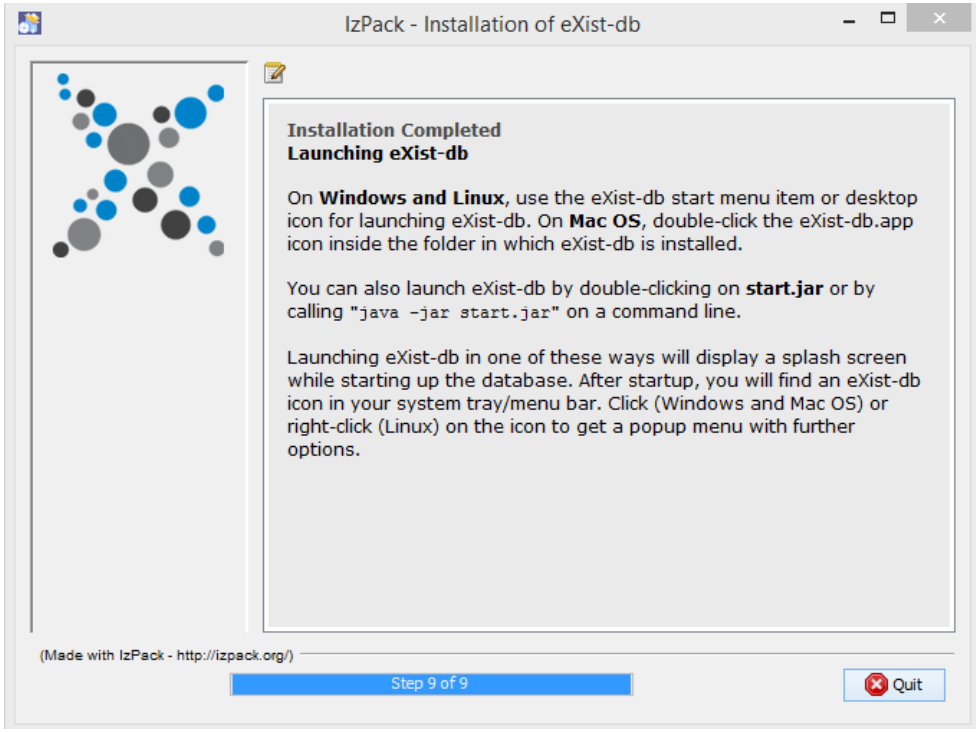
Instalación (V)





Instalación (VI)





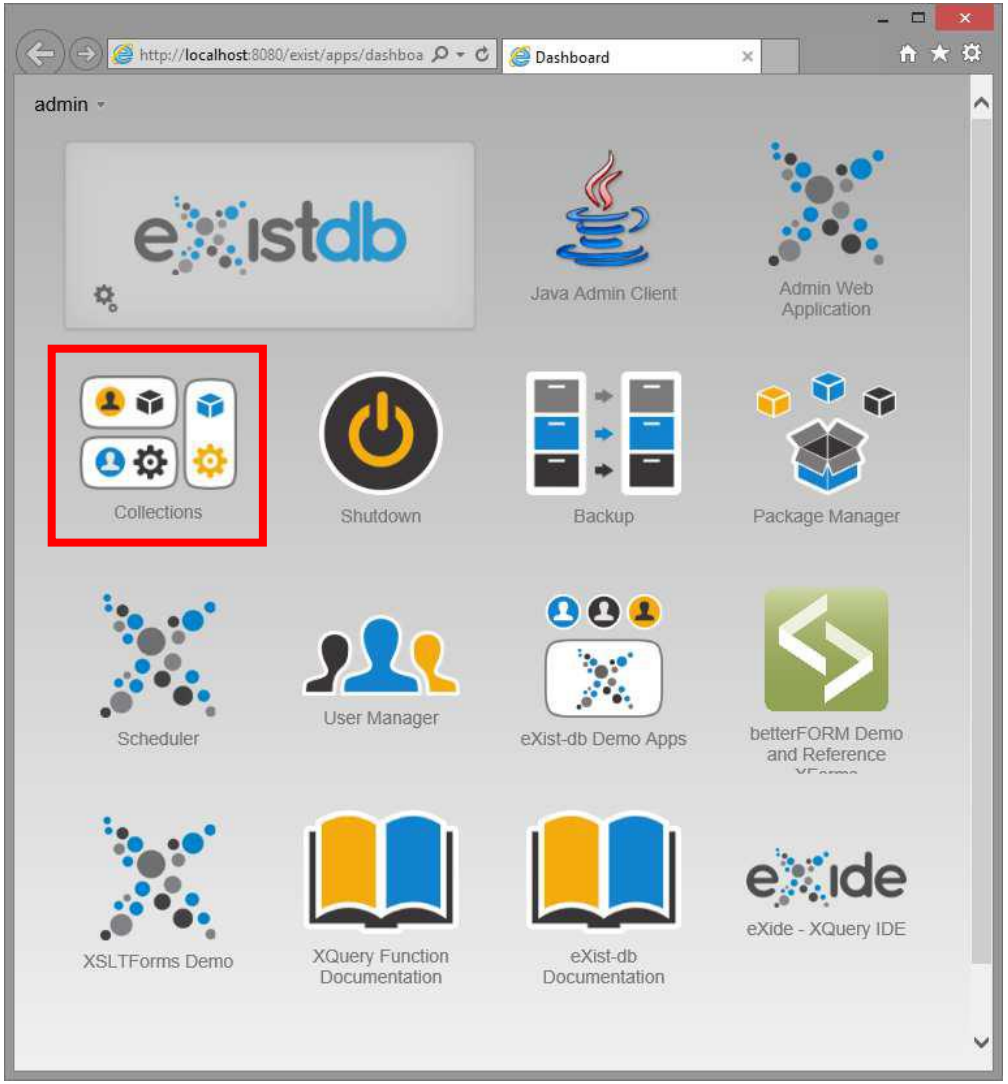
¡Ya tenemos eXist-db instalado!

Dashboard

- Para comenzar a emplear la base de datos es necesario crear **colecciones de documentos** e importar los ficheros a eXist

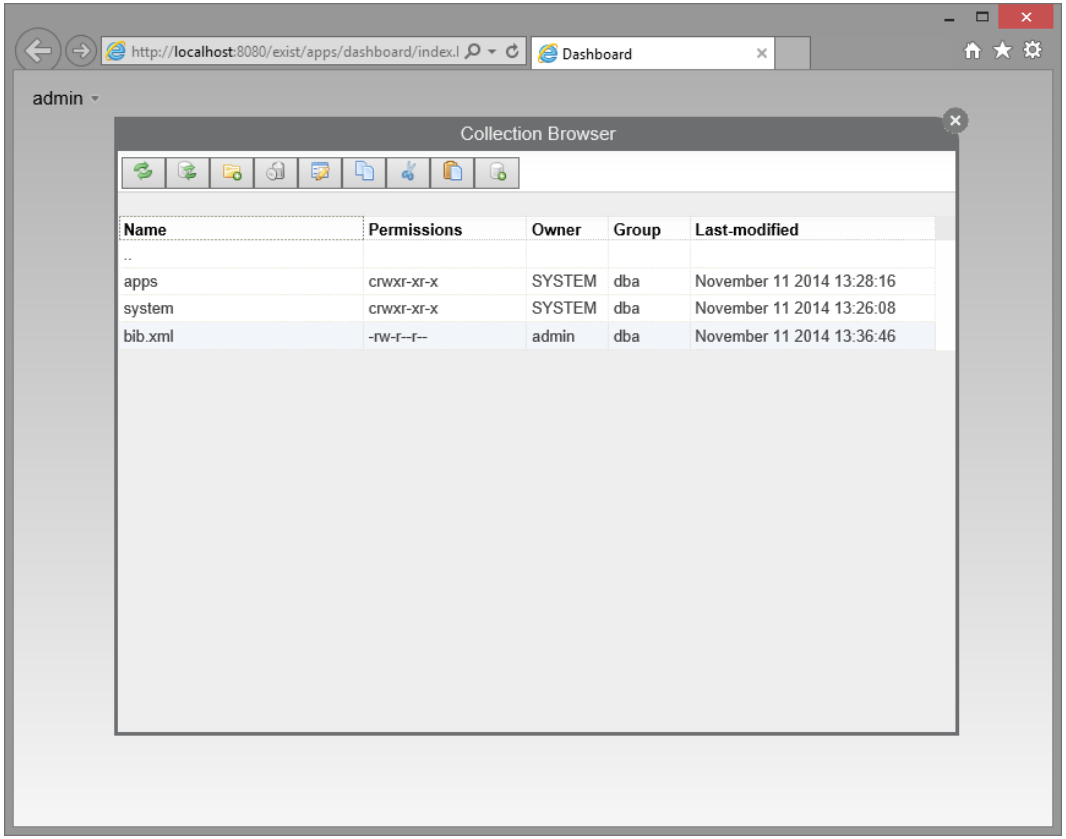
- Acceso
  - Usuario: admin
  - Password: admin

- Accedemos a *Collections*
  - Añadimos los ficheros con extensión xml



Dashboard

- Para comenzar a emplear la base de datos es necesario crear **colecciones de documentos** e importar los ficheros a eXist



- Acceso
  - Usuario: admin
  - Password: admin

- Accedemos a *Collections*
  - Añadimos los ficheros con extensión xml

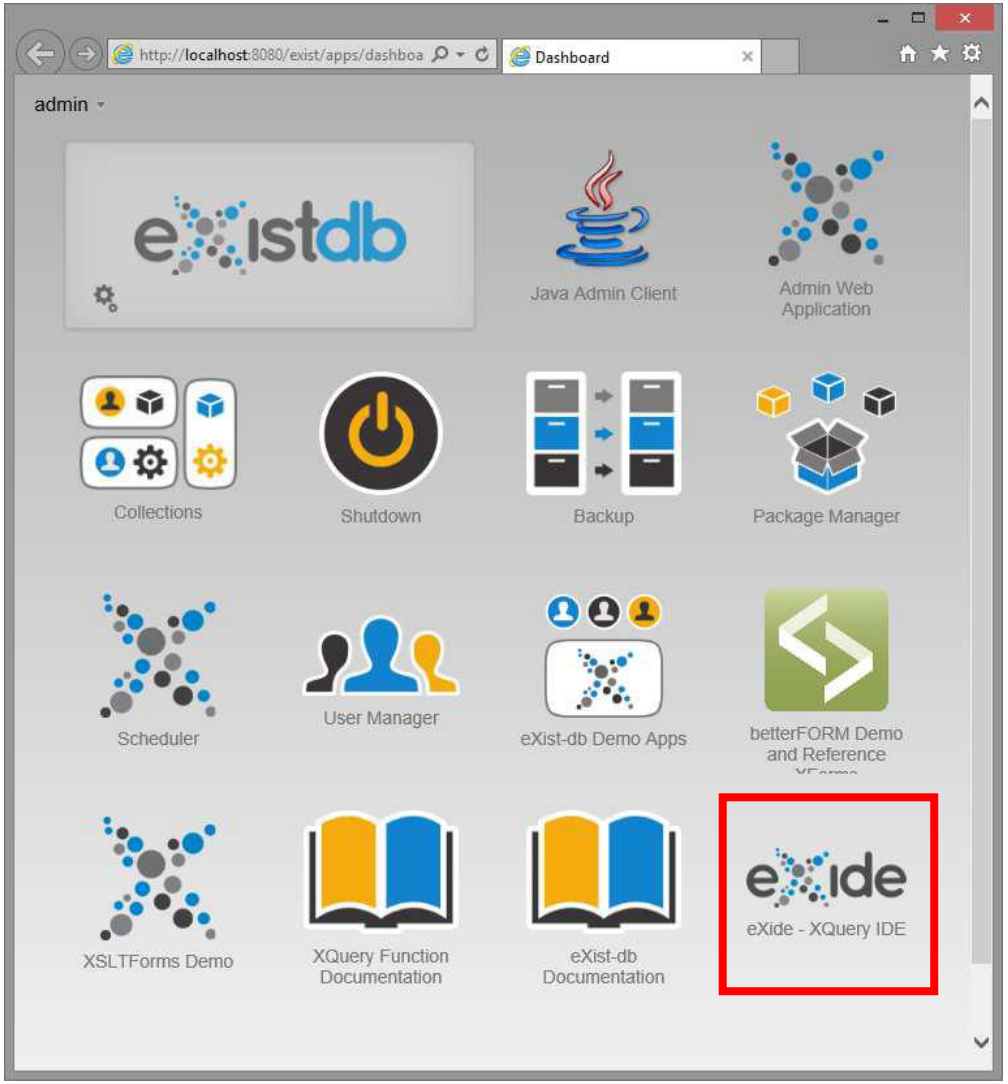
Dashboard

- Para comenzar a emplear la base de datos es necesario crear **colecciones de documentos** e importar los ficheros a eXist

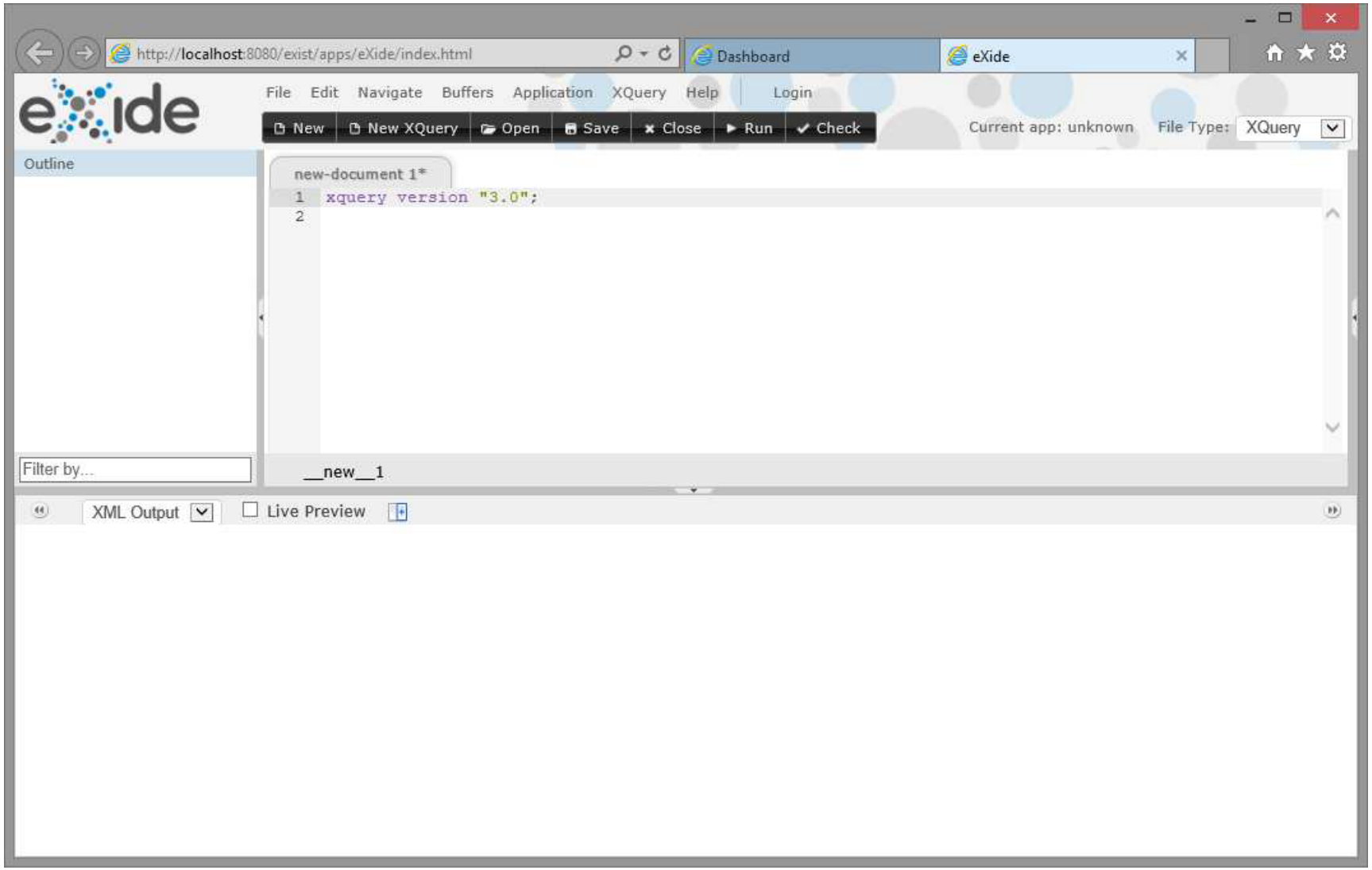
- Acceso
  - Usuario: admin
  - Password: admin

- Accedemos a *Collections*
  - Añadimos los ficheros con extensión xml

- Accedemos a eXide
  - XQuery IDE



Dashboard


http://localhost:8080/exist/apps/eXide/index.html
Dashboard
eXide
File Edit Navigate Buffers Application XQuery Help Login
New New XQuery Open Save Close Run Check
Current app: unknown File Type: XQuery
Outline
new-document 1*
1 xquery version "3.0";
2
Filter by...
__new__1
XML Output
Live Preview

Ejercicios para practicar

1. Obtener el titulo y el precio de los libros cuyo coste sea superior a 50 e inferior a 250

2. Obtener el título y el año de todos los libros publicados entre 1991 y 1999

3. Listar únicamente los libros que tienen exactamente un autor y cuyo año de publicación sea inferior a 2010

4. Listar todos los libros de bib.xml y un único autor para cada uno de ellos. Si el libro tiene más de un autor se añadirá la etiqueta <hay_mas> mostrando el número de autores que tiene el libro

¿Alguna Duda?