

Ficha de Trabalho nº 10

XQuery: Pesquisa e Transformação de documentos XML com XQuery

1. Introdução

Nesta ficha de trabalho pretende-se que os alunos explorem a construção de *queries* usando a sintaxe XQuery explorada nas aulas teóricas.

No Moodle encontram-se os ficheiros necessários à resolução desta ficha de trabalho.

2. Ficheiros

No Moodle disponibiliza-se o ficheiro que possui as funções necessárias ao tratamento dos ficheiros XQuery:

- **SaxonFunctions_XQuery**: funções que compilam e executam as queries em xQuery produzindo o `output` para XML, HTML ou TXT.
- Estas funções usam o API Saxon, já utilizado anteriormente para compilar as expressões XPath. O ficheiro **saxon9.jar** deve ser adicionado às Libraries do projecto Java.

No ficheiro **SaxonFunctions_XQuery.java** encontram-se três funções que permitem executar as *queries* e guardar o resultado num ficheiro:

- `public static void xQueryToText(String outputFile, String queryFile)`
- `public static void xQueryToHtml(String outputFile, String queryFile)`
- `public static void xQueryToXml(String outputFile, String queryFile)`

Explicação das funções:

As funções são semelhantes e apenas diferem no ficheiro output criado: a primeira gera um ficheiro de **texto**, a segunda um ficheiro **html** e a terceira um ficheiro **XML**.

Quando chamar a função no **main** deve enviar o nome do ficheiro .xquery onde implementou a query (**queryFile**) e o nome do ficheiro onde quer colocar o resultado produzido pela query (**outputFile**).

Os ficheiros XML de origem devem ser acedidos na query usando a instrução **doc("nomeficheiro")**

Por exemplo, no **main** pode escrever:

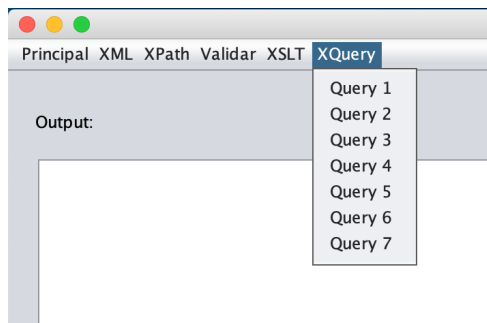
```
SaxonFunctions_XQuery.xQueryToText("resultado.txt", "livros1.xquery");
```

3. Exercícios

3.1 Alterar Projeto da aula anterior

Copie o ficheiro **SaxonFunctions_XQuery.java** para a pasta *src* do Projeto em desenvolvimento.

Na Janela de interface acrescente um novo menu XQuery com várias opções (vá acrescentando novos *MenuItem* para executar cada todas as queries indicadas a seguir):



As queries seguintes devem ser escritas em ficheiros com a extensão **.xql**. Pode usar um editor XML ou um editor de texto para escrever as queries. No Moodle encontra-se um *template* que pode usar para criar as queries. Grave os ficheiros na raiz do Projeto.

Nota: na função **doc** usada nas queries deve ser indicado o caminho todo do ficheiro XML
Por exemplo:

```
for $c in doc("C:/aulas/ID/Ficha10/livro.xml")//livro
```

O caminho onde se localiza o ficheiro não deve ter espaços nem caracteres acentuados ou outros.

3.2 Query 1: query1.xql

Crie um ficheiro XQL e grave-o com o nome **query1.xql**

Implemente uma expressão FLWOR para criar uma listagem TXT dos títulos dos livros cujo preço máximo seja superior ou igual a 10. O resultado deve ser enviado para um ficheiro de texto **livros1.txt** e deve ser ordenado pelo título.

Depois de implementada a query, no **evento da primeira opção do menu XQuery**, faça a chamada à função:

```
SaxonFunctions_XQuery.xQueryToText("livros1.txt", "query1.xql");
```

Complete o código da opção de menu para que após a Query ser executada apareça um Dialog informando que a operação foi feita com sucesso. Use a Classe **JOptionPane**:

```
JOptionPane.showMessageDialog(this,  
    "Query feita com sucesso... ficheiro TXT criado! ",  
    "XQuery",  
    JOptionPane.INFORMATION_MESSAGE);
```

Além disso, o resultado da query deve ser visualizado na *TextArea*: Faça como na Ficha anterior: use a Classe **Scanner** e **FileInputStream** para ler o ficheiro TXT criado e coloque o conteúdo na *TextArea*.

```
LIVRO: André e a Esfera Mágica -- 12.2  
LIVRO: Arsène Lupin - Cavaleiro Ladrão -- 15.5  
LIVRO: Cães Maus Não Dançam -- 13.41  
LIVRO: De Que Cor É um Beijinho? -- 11.9  
LIVRO: Ohsas 18001 -- 130.63  
LIVRO: Palácio da Lua -- 12.9
```

3.3 Query 2: query2.xql

Crie um ficheiro XQL e grave-o com o nome **query2.xql**

Implemente uma expressão FLWOR para criar um ficheiro HTML dos títulos correspondentes aos livros dos autores com apelido *Auster*. O resultado deve ser enviado para um ficheiro HTML **livros1.html** e deve ser ordenado pelo título.

Depois de implementada a query, na opção de Menu correspondente, faça a chamada à função:

SaxonFunctions_XQuery.xQueryToHtml("livros1.html", "query2.xql");

Complete o código da opção de menu para que após a Query ser executada apareça um Dialog informando que a operação foi feita com sucesso. Use a Classe **JOptionPane** como fez no exercício anterior.

Complete o código da opção de menu para que o resultado da query seja visualizado no browser.

Veja na **Ficha 9** o código para executar estes passos. O output deve ser semelhante ao mostrado abaixo:

Livros autor: Auster

- Homem na Escuridão
- Palácio da Lua

3.4 Query 3: query3.xql

Crie um ficheiro XQL e grave-o com o nome **query3.xql**

Crie uma query que use uma expressão FLWOR para criar um ficheiro XML dos títulos correspondentes aos livros com preço *bertrand* superior a 10 euros. A estrutura pretendida mostra-se na figura abaixo.

O resultado deve ser enviado para um ficheiro XML **livros1.xml** e deve ser ordenado pelo preço e como segundo critério de ordenação, o título.

Depois de implementada a query, na opção de Menu correspondente, faça a chamada à função:

SaxonFunctions_XQuery.xQueryToXml("livros1.xml", "query3.xql");

Abra o ficheiro **livros1.xml**, que deve ser semelhante a:

```
<queryResult>
  <livro>
    <titulo>De Que Cor É um Beijinho?</titulo>
    <autor>Rocio Bonilla</autor>
    <preco store="bertrand">11.9</preco>
  </livro>
  <livro>
    <titulo>André e a Esfera Mágica</titulo>
    <autor>Manuela Gonzaga</autor>
    <preco store="bertrand">12.2</preco>
  </livro>
  ...
</queryResult>
```

>> **Altere a query anterior para o valor do preço a estabelecer na filtragem do output seja introduzido pelo utilizador**

Para isso, crie um `jDialog` para receber o valor do utilizador e programe o botão do `jDialog`:

- O valor introduzido deve ser guardado num ficheiro XML de nome **valor.xml** com a seguinte estrutura `<valor>10</valor>` (10 seria o valor introduzido pelo utilizador). Se o ficheiro existir de execuções anteriores, deve ser criado de novo.

Para esta tarefa use as funções do JDOM

```
Element raiz = new Element("valor");
Document docV = new Document(raiz);
raiz.addContent(jTextField_valor.getText());
XMLJDomFunctions.escreverDocumentoParaFicheiro(docV, "valor.xml");
```

- Na query, para ler o valor introduzido pelo utilizador deve usar a função **doc** para o obter do ficheiro **valor.xml** e guarda-lo numa variável e use a função **number** para converter para número. No **where** substitua a constante pela variável `$v`

```
for ...
→ let $v := number(doc("C:/.../valor.xml")/valor)
  where ... > $v
  order by ...
  return ...
```

3.5 Query 4: livros4.xql

Nesta query pretende-se criar uma listagem HTML de todos os títulos vendidos pela BERTRAND com preços superiores a um valor introduzido pelo utilizador. O resultado deve aparecer ordenado por preço, e depois por título, como se mostra na figura abaixo.

Lista de Livros vendidos pela BERTRAND a preço superior a 12 euros

- André e a Esfera Mágica -- Preço BERTRAND: 12.2
- Arsène Lupin - Cavaleiro Ladrão -- Preço BERTRAND: 15.5
- Cães Maus Não Dançam -- Preço BERTRAND: 13.41
- Ohsas 18001 -- Preço BERTRAND: 130.63
- Palácio da Lua -- Preço BERTRAND: 12.9

**Valor introduzido
pelo utilizador**

3.6 Query 5: query5.xql

Crie uma query que, para cada autor, mostre os títulos dos seus livros. O resultado deve ser um ficheiro XML com a seguinte estrutura:

```
<TitulosAutor>
  <autor nome="Alice Vieira">
    <livro titulo="Leandro, Rei da Helíria"/>
  </autor>
  <autor nome="Antoine de Saint-Exupéry">
    <livro titulo="O Príncipezinho"/>
  </autor>
  <autor nome="Arturo Pérez-Reverte">
    <livro titulo="Cães Maus Não Dançam"/>
  </autor>
  ...
</TitulosAutor>
```

3.7 Query 6: query6.xql

Crie uma query que faça uma tabela HTML com as imagens de capa, títulos e autores dos livros, ordenado por título.

Catalogo de livros

Capa	Título	Autor
	André e a Esfera Mágica	Manuela Gonzaga
	Arsène Lupin - Cavaleiro Ladrão	Maurice Leblanc
	Cães Maus Não Dançam	Arturo Pérez-Reverte
	O Grande Gato	...

3.8 Query 7: query7.xql

Crie uma query que junte a informação do ficheiro **livro.xml** com a do ficheiro **livro_b.xml** (moodle) Itere todos os livros do primeiro ficheiro, se encontrar o mesmo isbn no segundo ficheiro junte a informação criando um ficheiro XML com a seguinte estrutura. O ficheiro está ordenado por valor de stock, nos stocks abaixo de 10 deve surgir um atributo adicional: **alerta="sim"**

```
<juntar>
  <livro isbn="9789722533492" stock="2" alerta="sim"/>
  <livro isbn="9789899039353" stock="3" alerta="sim"/>
  <livro isbn="9789722532877" stock="24"/>
  <livro isbn="9789722055932" stock="32"/>
  ...
</juntar>
```