

Documentação complementar:

- [XQuery 1.0/2.0 - Funções e Operadores](#)
- [XQuery 3.0 - Funções e Operadores](#)
- [Exemplos](#)

PARTE I

1. Considere o documento: `bookstore.xml`.
 - 1.1. Crie uma função XQuery que receba como parâmetro o nome de um autor e retorne apenas os livros desse autor;
 - 1.2. Crie uma função que receba como parâmetro uma categoria de livros e retorne a lista completa de livros se pelo menos um deles pertencer à categoria recebida por parâmetro (ver expressões quantificadas).
2. Utilizando a interface gráfica do BaseX crie uma base de dados: `bookstoreDB`. De seguida:
 - 2.1. Adicione dois documentos `BookStore.xml` (crie um segundo ficheiro com dados à sua escolha) à base de dados criada (Dica: utilize a função do BaseX: `db:add`).
 - 2.2. Realize as consultas do exercício 2 (Parte II) da ficha prática 4 considerando a base de dados criada (Dica: utilize a função do BaseX: `db:open`).
 - 2.3. Elimine um dos ficheiros introduzidos anteriormente na base de dados (Dica: utilize a função do BaseX: `db:delete`).

PARTE II

1. Considere os documentos XML: `authors` e `books`¹ (disponíveis na plataforma moodle).
 - 1.1. Escreva uma expressão XQuery que permita obter um novo documento com a informação do documento `authors` e `books`. Exemplo:

```
<books-complete-info>
  <book>
    <title>Structure and Interpretation of Computer Programs</title>
    <isbn>978-0-07-000422-1</isbn>
    <author-name>Harold Abelson</author-name>
  </book>
  <book>
    <title>Turtle Geometry</title>
    <isbn>978-0-262-01063-4</isbn>
    <author-name>Harold Abelson</author-name>
  </book>
  (...)
</books-complete-info>
```

1.2. Armazene o documento resultante numa base de dados BaseX.

1.3. Tendo por base a alínea 1.2., apresente uma transformação XQuery que através de um documento XML válido, apresente o número de autores de cada livro.

1.4. Tendo por base a alínea 1.2., apresente uma transformação XQuery que através de um documento XML válido, apresente o número de livros de cada autor (inclua o número de livros que não foi identificado o autor).

2. Considere os documentos: Fact1.xml e Fact2.xml que armazenam um conjunto de transações comerciais de uma dada empresa.

2.1. Armazene os documentos numa coleção em BaseX.

2.2. Apresente uma transformação XQuery que permita obter um resumo das taxas de IVA (elemento Tax) aplicadas em cada um dos produtos dos documentos envolvidos. Para cada taxa de IVA utilizada nos documentos indicados deverá ser criado um elemento “taxa”, onde:

- deverá representar o valor do IVA: `valor_iva`
- valor da soma dos preços de venda dos produtos sem o valor do IVA: `total_sem_iva`
- valor da soma (apenas) do valor do IVA: `total_iva`
- valor da soma dos preços de venda dos produtos com o valor do IVA: `total_com_iva`

Deverá ser apresentado um documento com a informação indicada tendo por base o seguinte vocabulário:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="resumo" type="resumoType" />

  <xs:complexType name="resumoType">
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="taxa" type="taxaType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="taxaType">
    <xs:sequence>
      <xs:element name="valor_iva" type="xs:float"/>
      <xs:element name="total_sem_iva" type="xs:float"/>
      <xs:element name="total_iva" type="xs:float"/>
      <xs:element name="total_com_iva" type="xs:float"/>
    </xs:sequence>
    <xs:attribute name="tipo" type="xs:string" />
    <xs:attribute name="codigo" type="xs:string" />
  </xs:complexType>
</xs:schema>
```

Exemplo de output:

```
<?xml version="1.0" encoding="UTF-8"?>
<resumo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="resumoIva.xsd">
  <taxa tipo="IVACON" codigo="RED">
    <valor_iva>5</valor_iva>
    <total_sem_iva>2.7</total_sem_iva>
    <total_iva>0.14</total_iva>
    <total_com_iva>2.84</total_com_iva>
  </taxa>
```

```
<taxa tipo="IVACON" codigo="Nor">
  <valor_iva>21</valor_iva>
  <total_sem_iva>2321.65</total_sem_iva>
  <total_iva>487.55</total_iva>
  <total_com_iva>2809.2</total_com_iva>
</taxa>
</resumo>
```

Notas:

- Crie uma função: `getResumoIVA` responsável por gerar cada elemento “taxa” para suportar o resumo de IVA.
- Os dados agregados são calculados a partir dos dois documentos disponibilizados.
- Valide o documento produzido utilizando a função `validate` (https://docs.basex.org/wiki/Validation_Module).