

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p><b>Processamento Estruturado de Informação</b></p> <p>1º Semestre</p> <p>Ficha Prática 5</p>
---	---

**Documentação complementar:**

- <https://www.w3.org/TR/xquery-operators/>
- <https://www.w3.org/TR/xpath-functions-30/>
- <http://www.datypic.com/books/xquery/chapter05.html>

1. Utilize o documento `bookstore.xml` para a resolução das seguintes alíneas:

- Crie uma função XQuery que receba como parâmetro o nome de um autor e que retorne todos os livros desse autor.
- Crie uma função XQuery que retorne todos os livros e que caracterize o estado de cada um deles em “antigo” ou “recente”. O estado “antigo” deverá corresponder aos livros publicados até 2010 (inclusive) e o estado “recente” aos livros posteriores a 2010. Para isso, inclua um elemento “state” em cada elemento “book”.
- Escreva uma expressão XQuery que permita determinar o número de livros que são anteriores a 2008.
- Escreva uma expressão quantificada que permita determinar se existem livros que custam menos do que 5.
- Escreva uma expressão XQuery que verifique se algum dos livros é da categoria “CHILDREN” e, em caso afirmativo, que retorne todos os livros da *bookstore*.
- Escreva uma expressão XQuery que verifique se todos os livros custam menos do que 60 e, em caso afirmativo, liste-os.

2. Crie uma diretoria `Preços` e coloque os ficheiros `precos_papelaria.xml` e `precos_informatica.xml` nessa diretoria. Construa expressões XQuery que permitam (dica: considere a função `collection`):

- Listar o nome dos produtos por ordem decrescente de preço.
- Indicar o preço médio dos produtos.
- Indicar o valor total a pagar por uma compra de 1 lápis e 2 borrachas.
- Listar o nome dos produtos que não têm desconto.

3. Considere o documento `voos.xml` fornecido na plataforma moodle. Escreva expressões XQuery que permitam realizar as seguintes consultas:

- Listar todos os diferentes países de onde são oriundos os passageiros
- Listar todos os voos que contêm "FR" no seu id do voo.

- c) Listar todos os voos que contêm "FR" no seu id do voo e que devolva o resultado com a seguinte estrutura:

```
<voo id="FR170">
  <origem>Frankfurt</origem>
  <destino>Zurich</destino>
</voo>
```

- d) Verificar se algum dos voos realizados no dia 2021-08-30 tinha como destino o aeroporto "NPL".
- e) Verificar se algum passageiro de nome "Santa Claus" viajou num voo nas condições indicadas na alínea anterior.
- f) Listar todos os voos realizados no dia 2021-08-30 com origem no aeroporto "NPL".
- g) Contar quantos voos cumprem as condições indicadas na alínea anterior.
- h) Listar o nome do passageiro que está associado ao passaporte com o número "000112".
- i) Listar o nome do passageiro que está associado à reserva com id igual a 1.
- j) Construir uma função que recebe como parâmetro o número de reserva e que devolva o nome do passageiro que lhe está associado.
- k) Construir uma função que deduza 10% ao valor da taxa de um dado aeroporto (considere a abreviatura como representativa do aeroporto).
- l) Aplicar um desconto de 10% ao valor da taxa de todos os aeroportos e retorne o novo valor da taxa para cada um dos aeroportos. Utilize a função definida anteriormente.

4. Utilizando BaseX, teste a seguinte transformação XQuery apresentada nos slides da aula TP:

```
element bookstore {
  for $book in http:send-request(<http:request method='get' />,
    'https://data.mongodb-api.com/app/data-
    docuz/endpoint/getBooks') [2]/json/_
  return element book {
    attribute category { $book/category },
    element title {
      $book/title/text()
    },
    if ($book/authors) then
      element authors {
        for $author in $book/authors/author
        return element author { $author/text() }
      }
    else
      element author { $book/author/text() },
      element year { $book/year/text() },
      element price { $book/price/text() }
  }
}
```