Nesta aula de laboratório iremos resolver exercícios envolvendo a manipulação de cadeias de caracteres através do uso de expressões regulares, ou a utilização de pesquisas com base em palavras-chave.

Tenha em atenção que a linguagem XPath 2.0 oferece de raiz algumas funções para a avaliação de expressões regulares, nomeadamente:

- Função `tokenize()`, que segmenta uma cadeia de caracteres, fornecida como primeiro parâmetro de entrada, numa sequência de cadeias de caracteres, usando como separador um caracter fornecido como segundo parâmetro de entrada.
- Função `matches()`, que retorna um valor Booleano com base no mapeamento entre uma cadeia de caracteres fornecida como primeiro parâmetro de entrada e uma expressão regular fornecida como segundo parâmetro de entrada.
- Função `replace()`, que pesquisa a cadeia de caracteres fornecida como primeiro parâmetro de entrada em todas as ocorrências da expressão regular fornecida como segundo parâmetro de entrada, substituindo todas as ocorrências pela cadeia de caracteres fornecida como terceiro parâmetro de entrada da função.

Como complemento a estas funções, o software usado nas aulas de laboratório inclui ainda suporte para uma extensão à linguagem XQuery relacionada com a pesquisa de texto em elementos XML (i.e., *XQuery with full-text search*).

No standard *XQuery with Full-Text*, **`ftcontains`** é o operador mais importante relacionado com a pesquisa por palavras chave. O operador retorna um valor Booleano no caso da pesquisa devolver nós XML que satisfaçam as condições indicadas pelo utilizador. O operador **`ftcontains`** pode ser usado em expressões XPath/XQuery com uma forma `search-domain ftcontains full-text-query`, tal como se exemplifica nos exemplos abaixo:

- Expressão que retorna os elementos de nome "element-name", cujo conteúdo textual contém a palavra-chave "keyword"

```
//element-name[ . ftcontains "keyword" ]
```

- Expressão FLWOR que retorna os elementos de nome "element-name", cujo conteúdo textual contém a palavra-chave "keyword", ordenados por relevância

```
for $hit score $score in //element-name[ . ftcontains "keyword" ]
order by $score descending
return $hit
```

O software usado nas aulas de laboratório inclui ainda algumas funções de extensão relacionadas com o processamento de informação textual:

- Função `java:gti.ftsentences()`, que retorna as frases (i.e., sequencias de palavras terminando num símbolo de pontuação) que se encontram definidas na cadeia de caracteres fornecida como parâmetro de entrada.

- Função `java:gti.textdoc()`, que aceita como parâmetro o URI de um documento e retorna a cadeia de caracteres correspondente ao seu conteúdo textual. Caso o parâmetro de entrada corresponda a um documento XML ou HTML, a função irá retornar apenas o conteúdo textual, removendo as *tags*.

Estas funções encontram-se codificadas em Java, podendo no entanto ser executadas no contexto de expressões XPath/XQuery.

Exercício 1

Considere o documento XML apresentado na Figura 1, o qual representa um conjunto de frases e informação sobre os seus autores das mesmas.

```
<?xml version="1.0"?>
<sentencesAndAuthors>
 <sentences>
  <p author="Richard Mutt">The milk costs $1.99.</p>
  <p author="Billy Shears">The newspaper is $1</p>
  <p author="Kelly Link">The newspaper costs 2.5€</p>
  <p author="Richard Mutt">Peanut butter is $2.49, and the candy bar is $0.65.</p>
  <p author="Richard Mutt">Milk is just too expensive.</p>
 </sentences>
 <authors>
  <info name="Richard Mutt">
    <infoLine1>30 Main St., New Haven, CT 06460</infoLine1>
    <infoLine2>Phone: 514-8888888</infoLine2>
  </info>

  <info name="Nanker Phelge">
    <infoLine1>1432 Milk St., Phoenix Arizona</infoLine1>
    <infoLine2>e-mail: nankerp@gmail.com</infoLine2>
  </info>

  <info name="Kelly Link">
    <infoLine1>1600 Pennsylvania Ave, Washington DC</infoLine1>
    <infoLine2>e-mail: kellylink@gmail.com</infoLine2>
  </info>

  <info name="Billy Shears">
    <infoLine1>1 Grand View Crest, Lansing, MI 22934-2234</infoLine1>
    <infoLine2>Email: bshears@hotmail.com Phonenumber: 224-88328181</infoLine2>
  </info>

  <info name="Richard Kelly">
    <infoLine1>15 Grand View Crest, Lansing, MI 22934-2244</infoLine1>
    <infoLine2>Phone: 224-88358488</infoLine2>
  </info>
 </authors>
</sentencesAndAuthors>
```

Figura 1: Documento XML usado nos exercícios de laboratório

Escreva expressões XQuery ou XPath que, utilizando as funções relacionadas com a avaliação de expressões regulares, permitam responder às seguintes questões:

1.1 - Para todos os autores que tenham um endereço de email definido, retornar o seu nome próprio (i.e. o primeiro nome) e o endereço de email correspondente.

1.2 - Para todos os autores que tenham um número de telefone definido, retornar o seu nome completo, a morada e o número de telefone correspondentes. A saída deverá ser apresentada segundo o formato exemplificado na Figura 3.

```
<autor>
 <nome>Nome do autor</nome>
 <telefone area="351">8888888</telefone>
 <street>1 Grand View Crest</street>
 <area>Lansing</area>
 <zip>MI 22934-2234</zip>
</autor>
```

Figura 3: Formato de saída para a XQuery da pergunta 1.2

Em relação ao número de telefone, a saída deverá indicar separadamente o código da área (i.e., o prefixo indicado em cada número de telefone). Em relação à morada, a saída deverá apresentar separadamente a rua, a área e o código postal.

1.3 - Retornar todos os valores monetários que são mencionados nas frases. No caso dos valores em dólares que são mencionados (e.g., $1.99), deverão ser convertidos para Euros usando a taxa de conversão 1.38 (e.g., o valor $1.99 deverá ser convertido para 1.44€).

Exercício 2

Para cada um dos problemas que se seguem, escreva uma função XQuery que aceite como parâmetro uma cadeia de caracteres e retorne um valor Booleano indicando se a cadeia de caracteres é válida no domínio do problema.

2.1 – Verificar se uma dada cadeia de caracteres corresponde a um endereço IP válido (i.e. uma sequencia de 4 valores numéricos compreendidos no intervalo [0-255] e separados entre si por um carácter “.”).

2.1 – Verificar se uma dada cadeia de caracteres corresponde a um URL bem formado, i.e. verificar se o URL respeita o formato que se apresenta abaixo:

```
protocol://hostname[:port][/path/filename][?param=value][&param2=value]
```

Considere que interessa apenas validar os URLs correspondentes a recursos acessíveis através dos protocolos HTTP, FTP, HTTPS ou FTPS.

2.2 – Verificar se uma dada cadeia de caracteres corresponde a uma data em português

de acordo com o formato “dia mês de ano” (e.g. “19 Novembro de 2008”). Tenha em atenção que o ano deverá ser superior a 1900. Deve ainda verificar se o valor numérico correspondente ao dia do mês é válido para o mês em questão, considerando que o mês de Fevereiro tem 29 dias.

2.3 – Verificar se uma dada cadeia de caracteres corresponde a uma palavra com pelo menos 5 caracteres alfabéticos, com inicio e fim numa vogal e apenas com consoantes nos caracteres intermédios (i.e., as vogais ocorrem apenas no início e no fim).

2.4 – Verificar se uma dada cadeia de caracteres corresponde a um valor no sistema numérico romano (i.e., XIX).

2.5 – Verificar se uma dada cadeia de caracteres contém apenas os algarismos 0 e 1 e se nela existe um número igual de ocorrências para as sub-cadeias “01” e “10”. Por exemplo:

- A cadeia '101' é válida pois contem uma ocorrência de '10' e outra de '01'.
- A cadeia '1001' é válida pois contem uma ocorrência de '10' e outra de '01'.
- A cadeia '1010' é inválida, pois contem duas ocorrências de '10' e apenas uma de '01'.
- A cadeia '10101' é válida pois contem duas ocorrências de '10' e duas de '01'.

Exercício 3

Com base no documento XML do Exercício 1, escreva expressões XQuery ou XPath que, utilizando os mecanismos relacionados com a pesquisa textual com base em palavras-chave, permitam responder às seguintes questões:

3.1 – Retornar o texto de todas as frases de autores cujo nome inclui a palavra “kelly” ou, em alternativa, a palavra “richard”.

3.2 – Encontrar todas as frases e todos as descrições de autores cujo texto (i.e., o conteúdo textual de todos os elementos descendentes) inclui a palavra “milk”.

3.3 – Encontrar todas as frases contendo a expressão “the milk costs”.

3.4 – Encontrar todas as frases de autores cujo nome contem a palavra “richard” e cujo texto da frase inclui um valor monetário. Na resolução desta alínea poderá combinar as funções relacionadas com a pesquisa textual e funções da biblioteca XPath 2.0 relacionadas com a avaliação de expressões regulares.

Exercício 4

Para cada um dos problemas que se seguem, escreva uma **função XQuery** que, utilizando as funções da biblioteca XPath 2.0 relacionadas com a **avaliação de expressões regulares**,

aceite como parâmetro uma cadeia de caracteres e retorne um valor Booleano indicando se a cadeia de caracteres é válida no domínio do problema.

4.1 - Verificar se uma dada cadeia de caracteres corresponde a um par de coordenadas de latitude e longitude, segundo o formato que se exemplifica abaixo.

```
25.03 Latitude; -9.5 Longitude
```

Tenha em atenção que a expressão regular deve verificar se os valores estão dentro dos intervalos válidos. A latitude varia entre -90.0000 e 90.0000 e a longitude varia entre -180.0000 e 180.0000. Os valores de latitude e longitude podem ainda ter um número indeterminado de 0s à esquerda da sua parte inteira. A parte decimal, caso esteja representada, ocupa um máximo de 4 algarismos.

4.2 – Verificar se uma dada cadeia de caracteres numéricos corresponde a uma capicua (i.e., uma cadeia que lida da direita para a esquerda ou da esquerda para a direita é idêntica) e se representa, simultaneamente, um número par com 6 dígitos.

Exercício 5

Os dois URLs indicados abaixo definem, respectivamente, uma colecção de textos representada em XML e um conjunto de pesquisas possíveis de ser efectuadas sobre a colecção de textos.

- http://dbappl.cs.utwente.nl/pftijah/data/cran.xml
- http://dbappl.cs.utwente.nl/pftijah/data/crantop.xml

1 - Escreva uma função XQuery que, para cada um dos tópicos de pesquisa do segundo documento XML, retorne os 10 textos mais relevantes tal como definidos no documento XML com a colecção. Note que poderá experimentar várias técnicas na realização da pesquisa *full-text* (e.g., procurar nos títulos ou não).

2 - Escreva uma expressão XQuery que, reutilizando a expressão da alínea anterior, retorne os 1000 textos mais relevantes no formato usado pela ferramenta *trec_eval* (ver o programa que se encontra em http://trec.nist.gov/trec_eval/)

Poderá avaliar a qualidade dos resultados obtidos pela sua pesquisa utilizando a ferramenta *trec_eval* com a informação sobre os "julgamentos de relevância" disponíveis para os vários pares tópico-texto -- http://dbappl.cs.utwente.nl/pftijah/data/cran.qrels.