Processamento Estruturado de Documentos 2001
LESI + LMCC (5º ano)

Ficha Teórico-Prática nº6

Data: 28 de Novembro de 2001

Exercício 1 (Submissão de TPs) *Foi montado um sistema para submissão electrónica de trabalhos práticos. Este sistema guarda a informação enviada num ficheiro XML com a seguinte estrutura:*

```
...
<grupo>
  <aluno1>
    <numero1>19670</numero1>
    <nome1>Alexandra Pinto</nome1>
    <curso1>LESI</curso1>
  </aluno1>
  <aluno2>
    <numero2>22639</numero2>
    <nome2>André Sousa</nome2>
    <curso2>LESI</curso2>
  </aluno2>
  <aluno3>
    <numero3></numero3>
    <nome3></nome3>
    <curso3></curso3>
  </aluno3>
  <titulo>Processador genérico de documentos estruturados (XML)</titulo>
  <file>19670_22639_.pli</file>
</grupo>
...
```

Agora é necessário criar várias stylesheets para processar esta informação e criar páginas HTML onde seja fácil consultar a informação.

1. *Especifique uma stylesheet que gera um novo ficheiro XML com a actualização do nome de algumas anotações: procederá à remoção do índice numérico nas etiquetas* aluno, numero, nome e curso.
2. *Especifique uma stylesheet para eliminar os campos vazios.*
3. *Especifique uma stylesheet que produz uma lista de alunos ordenada alfabeticamente por nome. Coloque a informação relativa a cada um dos alunos numa linha duma tabela HTML.*
4. *Especifique uma stylesheet que produz uma lista de alunos ordenada por número de aluno. Coloque a informação relativa a cada um dos alunos numa linha duma tabela HTML.*
5. *Especifique uma stylesheet que produz uma lista das submissões repetidas. Essa lista deverá ser um novo documento XML que deverá indicar para cada situação: a posição da submissão/grupo na árvore documental original, qual a situação anómala (o número que está repetido por exemplo); pense num DTD para este documento resultado.*

6. *Desenvolva uma stylesheet de nome* remove *que recebe um parâmetro numérico* n *e elimina o n-ésimo* grupo *do documento XML, o resultado é o mesmo documento XML mas sem aquela subárvore.*

7. *Desenvolva uma stylesheet de nome* consulta *que recebe um parâmetro numérico* n *e dá como resultado um novo documento (HTML?) com a informação respeitante ao grupo na posição n no documento original.*

8. *Desenvolva uma stylesheet de nome* take*, que recebe um parâmetro numérico* n*, e que dá como resultado um documento formado pelo conteúdo dos primeiros n grupos no documento original.*

9. *Desenvolva uma stylesheet de nome* drop*, que recebe um parâmetro numérico* n*, e que dá como resultado um documento formado pelo conteúdo dos últimos grupos do documento original após a eliminação dos primeiros n grupos.*

Exercício 2 (Bibliografia) *Descarregue o ficheiro relativo à Bibliografia do JCR da página da disciplina e tente resolver as alíneas seguintes:*

a) *Desenvolva uma stylesheet XSL que transforme aquele documento num outro com a seguinte estrutura:*

```
<bibtex>
  <entrada>
    <chave>atributo ID do original</chave>
    <titulo>...</titulo>
    <autores>
      <autor> nome completo do autor </autor>
      ...
    </autores>
    <data> ... </data>
    <outros-campos-que-sejam-necessários/>
  </entrada>
  ...
</bibtex>
```

b) *Especifique uma stylesheet XSL que garanta o invariante da chave: "não há chaves repetidas".*

c) *Especifique uma stylesheet XSL que processa o novo documento e produz um índice de autores que tem a seguinte estrutura:*

```
<ind-autor>
  <entrada>
    <autor>nome do autor</autor>
    <lchaves>
      <chave>chave dum livro escrito por este autor</chave>
      <chave>chave doutro livro escrito por este autor </chave>
      ... mais chaves ...
    </lchaves>
  </entrada>
  ... mais entradas, uma por cada autor ...
</ind-autor>
```

d) *Se ainda tiver tempo especifique um DTD para este tipo de documentos.*

Exercício 3 (Processando XML Schemas) *Na última aula teórica foram introduzidos os XML Schemas. Um XML Schema é um documento XML e como tal pode ser processado por uma stylesheet XSL.*

Considere o seguinte XML Schema relativo ao caso de estudo da agenda de contactos:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="agenda">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="entrada" type="Tentrada"/>
        <xs:element name="grupo" type="Tgrupo"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="Tentrada">
    <xs:sequence>
      <xs:element name="nome" type="xs:string"/>
      <xs:element name="eMail" type="xs:string"/>
      <xs:element name="telefone">
        <xs:simpleType>
          <xs:restriction base="xs:long">
            <xs:pattern value="[0-9]{9}"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
  </xs:complexType>
  <xs:complexType name="Tgrupo">
    <xs:choice maxOccurs="unbounded">
      <xs:element name="entrada" type="Tentrada"/>
      <xs:element name="grupo" type="Tgrupo"/>
      <xs:element name="referencia">
        <xs:complexType>
          <xs:attribute name="ref" type="xs:IDREF" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:choice>
    <xs:attribute name="gid" type="xs:ID" use="required"/>
  </xs:complexType>
</xs:schema>
```

a) *Especifique uma stylesheet XSL que faça uma lista dos elementos especificados no XML Schema.*

b) *Especifique uma stylesheet que calcula o grafo de dependências entre os elemntos definidos. O grafo gerado deverá ser representado por um documento XML do tipo:*

```
<grafo>
<elemento nome="nome-do-elemento">
  <composicao>
    <elemento nome="..."/>
    <elemento nome="..."/>
    ...
  </composicao>
```

```
</elemento>
...
</grafo>
```

c) *Tente, iterativamente, especificar uma stylesheet que gera automaticamente um DTD para o XML Schema processado. O DTD para o Schema apresentado deveria ser:*

```
<!ELEMENT agenda (entrada | grupo)+>
<!ELEMENT entrada (nome, eMail, telefone)>
<!ATTLIST entrada
  id ID #REQUIRED
>
<!ELEMENT grupo (entrada | grupo | referencia)+>
<!ATTLIST grupo
  gid ID #REQUIRED
>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT eMail (#PCDATA)>
<!ELEMENT telefone (#PCDATA)>
<!ELEMENT entrada (nome, eMail, telefone)>
<!ATTLIST entrada
  id ID #REQUIRED
>
<!ELEMENT grupo (entrada | grupo | referencia)+>
<!ATTLIST grupo
  gid ID #REQUIRED
>
<!ELEMENT referencia EMPTY>
<!ATTLIST referencia
  ref IDREF #REQUIRED
>
```