



Universidade Federal de Viçosa - *Campus* Florestal
Bacharelado em Ciência da Computação
CCF 492 - Tópicos Especiais II
Prof. Daniel Mendes Barbosa

Trabalho 01

Dados semi-estruturados: uso de XSLT

Samuel Jhonata S. Tavares	2282
Wandella Maia de Oliveira	2292

Florestal - MG
2018

Sumário

1	INTRODUÇÃO	3
2	DESENVOLVIMENTO	4
2.1	Considerações Gerais	4
2.2	Explicações	4
2.2.1	XML	4
2.2.2	XSL	4
2.2.3	XSLT	5
2.3	Exemplo	5
2.3.1	Dados utilizados	5
2.3.2	Criação XSL	7
2.3.3	Transformação com XSLT	9
2.3.4	Resultado	11
3	CONCLUSÃO	13
	REFERÊNCIAS	14

1 Introdução

Este trabalho tem por objetivo apresentar com mais detalhes a *XSLT* (*eXtensible Stylesheet Language for Transformation*), a fim de complementar o conteúdo visto em sala sobre dados semi-estruturados, utilizando como exemplo dados do Governo Federal do Brasil sobre fornecedores de materiais e serviços, disponibilizadas através de um arquivo *XML* no portal de dados.

No capítulo 2, é apresentado todo o desenvolvimento do trabalho, com suas considerações gerais na seção 2.1, seguidas das explicações sobre *XML*, *XSL* e *XSLT* na seção 2.2 e um exemplo de aplicação na seção 2.3. Já no capítulo 3, é apresentada uma breve conclusão.

2 Desenvolvimento

2.1 Considerações Gerais

Este trabalho utiliza dados do Ministério do Planejamento do Brasil (GOVERNO FEDERAL DO BRASIL, 2014) com informações sobre compras públicas do Governo Federal. Buscou-se apresentar um breve resumo sobre XML, XSL e XSLT e, através do uso dos dados mencionados, criar um exemplo de uma transformação XSLT.

Como ferramenta, foi utilizada a *IDE Netbeans* para gerar o *DTD* do *XML*, a criação do arquivo *XSL* e a transformação *XSLT* em *HTML*. Para a interpretação do *HTML* gerado, foi utilizado o *browser Google Chrome*.

2.2 Explicações

2.2.1 XML

XML(eXtensible Markup Language) é um padrão proposto pela *W3C(World Wide Web Consortium)* para estruturar informações da web através de marcadores na estruturação, assim como o *HTML* (W3SCHOOLS, 2018). O *XML* é um subconjunto do *SGML(Standard Generalized Markup Language)* que especifica regras para linguagens de marcação independente da plataforma.

O objetivo do *XML* é criar documentos que oferece dinamismo para organizar os dados e isso é possível através de suas *tags* para possibilitar a separação dos conteúdos (NASCIMENTO, 2001). Tais *tags* irão armazenar os documentos estruturados em texto puro e renderizá-los em grande variedade de modos. Além disso, para formatar a apresentação dos dados, é necessário usar *XSL*, que será explicado no próximo tópico.

2.2.2 XSL

O *XSL(XML Extensible Stylesheet Language)* é uma linguagem para usar folhas de estilo ou *Stylesheet*(W3SCHOOLS, 2018). Ao usar o *XML* estruturado, é usado o *XSL* com *stylesheet* para expressar o conteúdo do arquivo que será apresentado e, para visualizar o resultado, pode-se utilizar um *browser* compatível.

A formatação do *XSL* é obtida através de *Stylesheet* e é feito por um processador. Este software que irá processar o documento *XML* e um *XSL*, irá produzir uma apresentação dos dados que estarão armazenados no *XML*. Para compreender o resultado final, é necessário entender dois passos que explicam melhor a transformação. Primeiro,

o *XML* é interpretado como uma árvore, onde os metadados estão armazenados e o segundo, é a interpretação da árvore para produzir resultados ajustados a mídia desejada (NASCIMENTO, 2001).

Vale ressaltar que todo documento *XML* tem sua organização como árvore. O documento possui seu início na raiz e outros atributos são tidos como filhos, descendo a hierarquia da árvore e dependendo do contexto da aplicação dos dados semi-estruturados, como depois será mostrado na hierarquia da Figura 2.

2.2.3 XSLT

A *XSLT* é recomendada para definir a transformação e apresentação de documentos no formato *XML*, sendo uma linguagem de transformação de documentos desse formato para ele mesmo, para formato de texto ou *HTML* (W3C, 2015). Ela pode ser usada utilizada de forma independente do *XSL*, porém foi desenvolvida para trabalharem em conjunto (NASCIMENTO, 2001).

A *XSLT* utiliza *templates* para as transformações através do uso de regras sobre uma árvore fonte *XML*, gerando uma árvore resultado. Uma regra de *template* é formada por um padrão que é comparado com a árvore fonte e um *template* que gera a árvore resultado, possibilitando ser usado em diversos documentos que trabalhem com estruturas de árvores semelhantes. (NASCIMENTO, 2001)

2.3 Exemplo

2.3.1 Dados utilizados

Os dados utilizados neste trabalho foram retirados da página do Governo Federal do Brasil e têm informações sobre pessoas físicas e jurídicas que fornecem materiais e serviços ao governo.

A única modificação realizada nos dados foi a substituição, nos nomes das empresas/pessoas, do símbolo & por 'e', para que fosse possível a criação do *DTD*.

Como é possível ver na Figura 3, onde é apresentado o *DTD* do conjunto de dados, o elemento *resource*, que representa uma instância do fornecedor, é composta de atributos como *id*, *cnpj*, *cpf*, *nome*, *uf*, entre outros.

Figura 1 – DTD do conjunto de dados utilizados

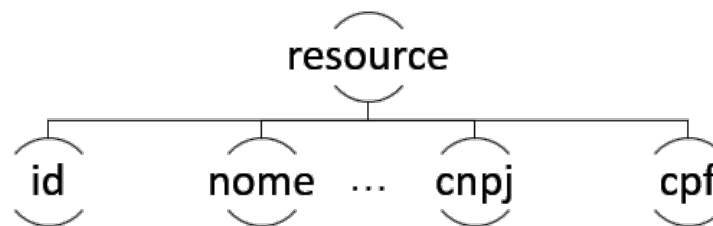
```

1 <?xml version='1.0' encoding='UTF-8'?>
2
3 <ELEMENT resource (id|_links|_embedded|cnpj|nome|ativo|recadastrado|habilitado|licitar|id_ramo_negocio|id_cnae|id_municipio|uf|id_unidade_cadastradora|cpf|id_natureza_juridica|id_porte_empresa)>
4 <_ATTLIST resource
5   count CDATA #IMPLIED
6   offset CDATA #IMPLIED
7 >
8
9 <ELEMENT _links (link)*>
10
11 <ELEMENT link EMPTY>
12 <_ATTLIST link
13   href CDATA #IMPLIED
14   rel CDATA #IMPLIED
15   title CDATA #IMPLIED
16 >
17
18 <ELEMENT _embedded (resource)*>
19 <ELEMENT id (#PCDATA)>
20 <ELEMENT cnpj (#PCDATA)>
21 <ELEMENT nome (#PCDATA)>
22 <ELEMENT ativo (#PCDATA)>
23 <ELEMENT recadastrado (#PCDATA)>
24 <ELEMENT id_municipio (#PCDATA)>
25 <ELEMENT uf (#PCDATA)>
26 <ELEMENT id_natureza_juridica (#PCDATA)>
27 <ELEMENT id_porte_empresa (#PCDATA)>
28 <ELEMENT id_ramo_negocio (#PCDATA)>
29 <ELEMENT id_unidade_cadastradora (#PCDATA)>
30 <ELEMENT id_cnae (#PCDATA)>
31 <ELEMENT habilitado_licitar (#PCDATA)>
32 <ELEMENT cpf (#PCDATA)>

```

Na Figura 2 é mostrada uma parte da hierarquia de dados do conjunto utilizado.

Figura 2 – Hierarquia dos dados do conjunto utilizado



Na Figura 3 é mostrado uma parte do arquivo *XML* retirado do site, com diversos registros de fornecedores.

Figura 3 – Parte do XML do conjunto de dados utilizados

```

1 <resource count="391825" offset="0">
2   <_links>
3     <link href="http://compras.dados.gov.br/fornecedores/vl/fornecedores.xml" rel="self" title="Lista de fornecedores"/>
4     <link href="/fornecedores/vl/fornecedores.xml?offset=0" rel="first" title="Primeira página"/>
5     <link href="/fornecedores/vl/fornecedores.xml?offset=500" rel="next" title="Próxima página"/>
6   </_links>
7   <_embedded>
8     <resource>
9       <_links>
10        <link href="/fornecedores/id/fornecedor_pj/00000000000191" rel="self" title="Fornecedor PJ 00.000.000/0001-91: BANCO DO BRASIL SA"/>
11        <link href="/fornecedores/id/municipio/97012" rel="municipio" title="Município 97012: BRASÍLIA"/>
12        <link href="/licitacoes/id/uasg/130005" rel="uasg" title="UASG 130005: COORD.-GERAL DE EXECUCAO ORÇ.E FIN./DA/MAPA"/>
13        <link href="/fornecedores/id/natureza_juridica/9" rel="natureza_juridica" title="Natureza Jurídica 9: SOCIEDADE DE ECONOMIA MISTA"/>
14        <link href="/fornecedores/id/ramo_negocio/94" rel="ramo_negocio" title="Ramo de Negócio 94: ATIVIDADES DE ORGANIZAÇÕES ASSOCIATIVAS"/>
15        <link href="/fornecedores/id/porte_empresa/5" rel="porte_empresa" title="Porte da Empresa 5: DEMAIS"/>
16        <link href="/fornecedores/id/cnae/6422100" rel="cnae" title="CNAE 6422100: BANCOS MÚLTIPLOS, COM CARTEIRA COMERCIAL"/>
17      </_links>
18      <id>1</id>
19      <cnpj>000000000000191</cnpj>
20      <nome>BANCO DO BRASIL SA</nome>
21      <ativo>true</ativo>
22      <recadastrado>true</recadastrado>
23      <id_municipio>97012</id_municipio>
24      <uf>DF</uf>
25      <id_natureza_juridica>9</id_natureza_juridica>
26      <id_porte_empresa>5</id_porte_empresa>
27      <id_ramo_negocio>94</id_ramo_negocio>
28      <id_unidade_cadastradora>130005</id_unidade_cadastradora>
29      <id_cnae>6422100</id_cnae>
30      <habilitado_licitar>true</habilitado_licitar>
31    </resource>

```

2.3.2 Criação XSL

Foi criado um arquivo *XSL* com o nome **fornecedores.xsl** com o auxílio da *IDE Netbeans*, onde foram inseridos comandos *HTML* e também *XSL*, como é possível notar nas Figuras 4, 5 e 6, que apresentam trechos do código onde é criado o menu, a apresentação das pessoas jurídicas e a apresentação das pessoas físicas, respectivamente.

Figura 4 – Documento XSL criado - menu

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
4    <xsl:output method="html"/>
5
6    <xsl:template match="/">
7      <html>
8        <head>
9          <title>Fornecedores</title>
10         <link rel="stylesheet" type="text/css" href="estilo.css"/>
11         <script type="text/javascript" src="funcoes.js"/>
12       </head>
13
14       <body>
15         <div>
16           <h1>Fornecedores do Governo Federal do Brasil </h1>
17         </div>
18
19         <div class="menu">
20           <div class="btn_menu">
21             <button onclick="mostraEsconde('empresas', 1); mostraEsconde('pf', 0)">Pessoa Juridica</button>
22           </div>
23           <div class="btn_menu">
24             <button onclick="mostraEsconde('empresas', 0); mostraEsconde('pf', 1)">Pessoa Fisica</button>
25           </div>
26         </div>
27
28       </body>
29     </template>
30   </xsl:stylesheet>
```

Figura 5 – Documento XSL criado - empresas

```
30 <!-- empresas -->
31 <div id="empresas">
32   <table border="1">
33     <tr>
34       <th>Empresa</th>
35       <th>CNPJ</th>
36       <th>UF</th>
37     </tr>
38     <xsl:for-each select="resource/_embedded/resource">
39       <xsl:if test="cnpj != ''">
40         <tr>
41           <td>
42             <xsl:value-of select="nome" />
43           </td>
44           <td>
45             <xsl:value-of select="cnpj" />
46           </td>
47           <td>
48             <xsl:value-of select="uf" />
49           </td>
50         </tr>
51       </xsl:if>
52     </xsl:for-each>
53   </table>
54 </div>
55
56 <br/>
```

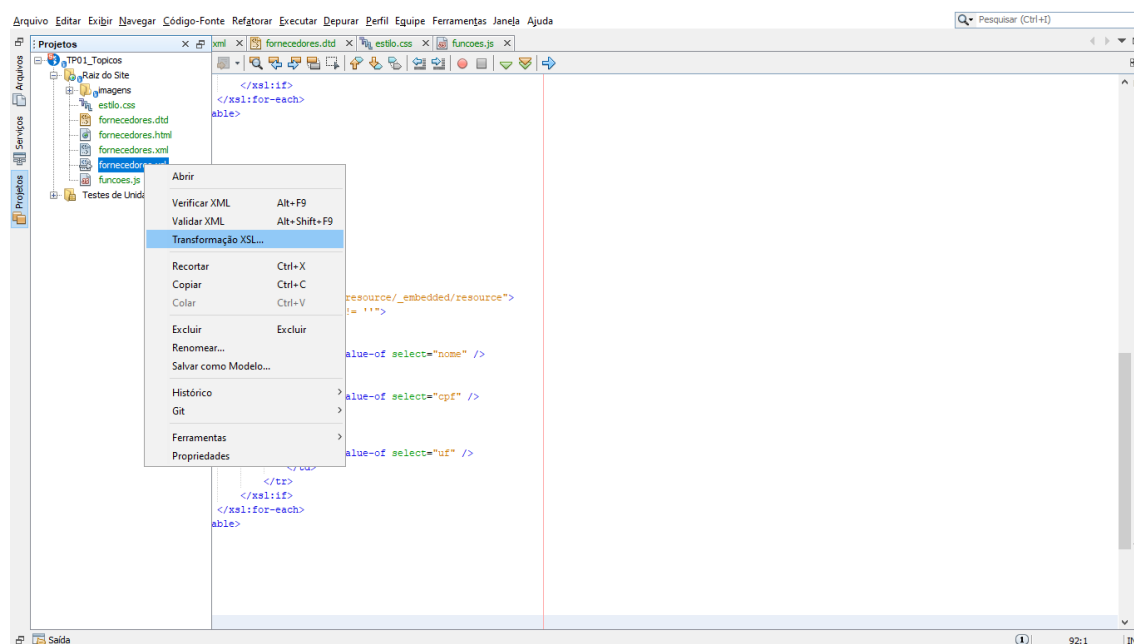
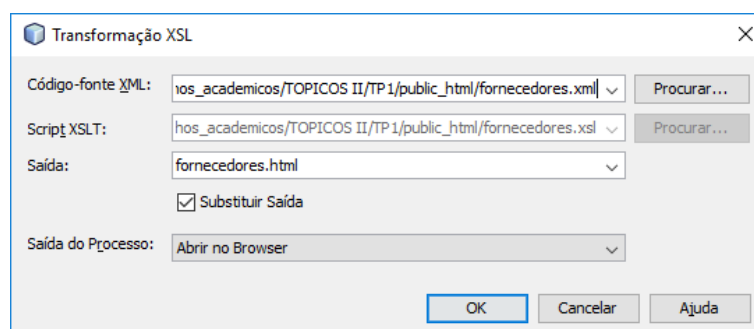

Figura 6 – Documento XSL criado - pessoas físicas

```
61 <!-- pessoas fisicas -->
62 <div id="pf">
63   <table border="1">
64     <tr>
65       <th>Nome</th>
66       <th>CPF</th>
67       <th>UF</th>
68     </tr>
69     <xsl:for-each select="resource/_embedded/resource">
70       <xsl:if test="cpf != ''">
71         <tr>
72           <td>
73             <xsl:value-of select="nome" />
74           </td>
75           <td>
76             <xsl:value-of select="cpf" />
77           </td>
78           <td>
79             <xsl:value-of select="uf" />
80           </td>
81         </tr>
82       </xsl:if>
83     </xsl:for-each>
84   </table>
85 </div>
86 </body>
87 </html>
88 </xsl:template>
89
90
91 </xsl:stylesheet>
```

Também foram criados dois arquivos para o estilo *CSS*, *estilo.css*, e funções *Javascript*, *funcoes.js*, que foram utilizadas em *fornecedores.html*, aos quais não serão detalhados nesse relatório.

2.3.3 Transformação com XSLT

Com o código *XSL* criado, com o auxílio da *IDE Netbeans*, foi feita a transformação *XSLT*, como é mostrado nas Figuras 7 e 8.

Figura 7 – Executando a transformação *XSLT* com auxílio da *IDE Netbeans* - parte 1Figura 8 – Executando a transformação *XSLT* com auxílio da *IDE Netbeans* - parte 2

Após a transformação, foi gerado o arquivo *fornecedores.html*, que é mostrado em partes na Figura 9.

Figura 9 – Documento HTML gerado através da transformação XSLT

```

1 <html>
2 <head>
3 <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 <title>Fornecedores</title>
5 <link rel="stylesheet" type="text/css" href="estilo.css">
6 <script type="text/javascript" src="funcoes.js"></script>
7 </head>
8 <body>
9 <div>
10 <h1>Fornecedores do Governo Federal do Brasil </h1>
11 </div>
12 <div class="menu">
13 <div class="btn_menu">
14 <button onclick="mostraEsconde('empresas', 1); mostraEsconde('pf', 0)">Pessoa Juridica</button>
15 </div>
16 <div class="btn_menu">
17 <button onclick="mostraEsconde('empresas', 0); mostraEsconde('pf', 1)">Pessoa Fisica</button>
18 </div>
19 </div>
20 <div id="empresas">
21 <table border="1">
22 <tr>
23 <th>Empresa</th><th>CNPJ</th><th>UF</th>
24 </tr>
25 <tr>
26 <td>BANCO DO BRASIL SA</td><td>000000000000191</td><td>DF</td>
27 </tr>
28 <tr>
29 <td>BANCO DO BRASIL SA</td><td>000000000016276</td><td>MG</td>
30 </tr>
31 <tr>
32 <td>BANCO DO BRASIL SA</td><td>000000000425109</td><td>DF</td>
33 </tr>
34 <tr>
35 <td>TARGET ENGENHARIA E CONSULTORIA LTDA.</td><td>000000028000129</td><td>SP</td>
36 </tr>
37 <tr>
38 <td>COUPLING LTDA</td><td>00000155000128</td><td>SP</td>

```

2.3.4 Resultado

A seguir, nas figuras 10, 11 e 12 é possível ver as páginas carregadas no navegador *Google Chrome*, com a página esperando a solicitação do usuário, após o usuário escolher a opção "*Pessoa Jurídica*" e após o usuário escolher a opção "*Pessoa Física*", respectivamente.

Figura 10 – Página aberta no navegador



Figura 11 – Página aberta no navegador - Pessoa Jurídica



Empresa	CNPJ	UF
BANCO DO BRASIL SA	00000000000191	DF
BANCO DO BRASIL SA	00000000016276	MG
BANCO DO BRASIL SA	00000000425109	DF
TARGET ENGENHARIA E CONSULTORIA LTDA	00000028000129	SP
COUPLING LTDA	00000155000128	SP
IRMAOS RODOPOULOS LTDA	00000166000108	DF
IRMAOS RODOPOULOS LTDA	00000166000299	DF
BRB BANCO DE BRASILIA SA	00000208000100	DF
C.C. ENGENHARIA CONSULTIVA E PROJETOS DE AR CONDICIONADO	00000341000167	SP
BIOAGRI ANALISES DE ALIMENTOS LTDA	00000410000132	SP
DEBRITO PROPAGANDA LTDA	00000424000156	SP
DEBRITO PROPAGANDA LTDA	00000424000318	DF
CASA DOS PARAFUSOS LTDA	00000463000153	DF
HIC PNEUS S/A	00000802001769	PE
BRASAL-BRASILIA SERVICOS AUTOMOTORES S/A	00000885000129	DF
SUPERMERCADO SERVE TODOS PIRAJUI LTDA	00000993000100	SP
SUPERMERCADO SERVE TODOS PIRAJUI LTDA	00000993000363	SP
SUPERMERCADO SERVE TODOS PIRAJUI LTDA	00000993000444	SP
SUPERMERCADO SERVE TODOS PIRAJUI LTDA	00000993000525	SP
SUPERMERCADO SERVE TODOS PIRAJUI LTDA	00000993000606	SP
SA CORREIO BRAZILIENSE	00001172000180	DF

Figura 12 – Página aberta no navegador - Pessoa Física



Nome	CPF	UF
VANDERLEI BATALHA DOS SANTOS FILHO	***007562**	AM
ISABELLE SANTOS SOARES	***008237**	RJ
JULIANO DERLIV EICHELT	***012001**	MT
JOSE LUIZ MOTA MENEZES	***012604**	PE
FABIO ANTONIO DEZAN	***026841**	MT
DANIEL LOBO FERNANDES	***030487**	SC
FRANCISCO LEMOS SANT'ANA	***032045**	BA
ANA BEATRIZ FERNANDES CERBINO	***034287**	RJ
ROMULO PASCOAL PEDROSO	***048720**	RS
OCTAVIO PINTO CARVALHEIRA	***050964**	PE
ANDRE SELBACH NASI	***054400**	RS
VIRGILIO JOSE LEMOS ABREU	***062891**	MG
MARIA AURENICE FERREIRA DE ABREU	***072603**	CE
JEAN CARLOS DE SOUZA	***073321**	DF
HELENA MITIE OSUGUI MANTOVANI	***074268**	MT
EVANILDO CAVALCANTE BECHARA	***074609**	RJ
MANOEL ALFREDO CAMARAO DE ALBUQUERQUE	***080951**	MG
ANGELA DE LIMA MACHADO	***098400**	SC
HARCO OFUGI RODRIGUES	***098431**	DF
ANTONIO MARCOS CRUZ DOS SANTOS	***100833**	CE
INIVA TEREZINHA BIZZO VITTI MA	***105170**	RS

3 Conclusão

Com esse trabalho foi possível aprender, na prática, como trabalhar com dados semi-estruturados em *XML*, utilizando XSL e a transformação *XSLT* para a elaboração de um *template*, tendo como saída um arquivo *HTML*, além de utilizar *JavaScript* e *CSS*.

Referências

GOVERNO FEDERAL DO BRASIL. *Compras públicas do governo federal*. 2014.

Disponível em: <<http://dados.gov.br/dataset/compras-publicas-do-governo-federal>>.

Acesso em: 28 ago. 2018.

NASCIMENTO, A. E. K. Intercâmbio de dados entre aplicativos utilizando xml/xslt. 2001.

UFCG. *Uso de XML/XSLT*. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/daca/html/xsl/index.html>>. Acesso em: 28 ago. 2018.

W3C. *TRANSFORMATION*. 2015. Disponível em: <<https://www.w3.org/standards/xml/transformation>>. Acesso em: 28 ago. 2018.

W3SCHOOLS. *XML Tutorial*. 2018. Disponível em: <<https://www.w3schools.com/xml>>. Acesso em: 28 ago. 2018.