

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Processamento Estruturado de Documentos

Ano Lectivo de 2001/2002



escola de engenharia



departamento de
informática

Transformação XML2HTML

Carlos Carvalho, CEI n.º 5337

Julho, 2002

Data de Recepção	
Responsável	
Avaliação	
Observações	

Transformação XML2HTML

Carlos Carvalho, CEI n.º 5337

Julho. 2002

A todos que têm aturado a minha falta
de comparência este ano...

Especialmente à minha família:
esposa e filhos, que nunca deixaram
de estar ao meu lado.

Agradecimentos

Agradeço aos meus colegas de curso e escola, pelas suas preciosas e pertinentes opiniões, à minha família, por saber estar lá (e não estar...) quando foi preciso.

Resumo

Com o presente trabalho pretende-se por em prática um conjunto de conceitos e ferramentas no âmbito da documentação estrutural. A ideia original, conforme enunciada, constituía na utilização da pré-tese como base de trabalho. Contudo, o facto daquele documento ainda não se encontrar minimamente desenvolvido, torna-o inviável. Desta forma este mesmo relatório do projecto serviu como suporte sobre o qual desenvolvi: um documento XML, um DTD correspondente, um *schema* e uma transformação XSL para transformação do texto XML numa página HTML.

Área de Aplicação: Estruturação e processamento de documentos em XML

Palavras-Chave: Documento estruturado, anotação, XML, DTD, Schemas, XPath, XSLT, XSL, HTML

Índice Remissivo

1.	Introdução	4
1.1.	Contextualização	4
1.2.	Apresentação do caso de estudo	4
1.3.	Motivação e Objectivos	4
1.4.	Estrutura do Relatório	5
2.	O documento XML	6
2.1.	A estrutura em DTD	8
2.2.	O Schema do documento	9
3.	A transformação XML2HTML	11
4.	Conclusões e trabalho futuro	14
	Bibliografia	15
Anexos		
Anexo 1	Estrutura DTD	16
Anexo 2	Estrutura <i>Schema</i>	17
Anexo 3	Transformação XSL	20

1. Introdução

1.1. Contextualização

O projecto a que o presente respeita, insere-se no âmbito da disciplina de Processamento Estruturado de Documentos do Mestrado/Cursos de Especialização em Informática, servindo como elemento de avaliação opcional da referida disciplina.

1.2. Apresentação do caso de estudo

De acordo com o enunciado expresso, pretendia-se elaborar um documento em *eXtended Markup Language* (XML), recorrendo à pré-tese do mestrado como suporte do mesmo. Para além deste, constava do enunciado a criação de um *Document Type Definition* (DTD) e de um *Schema* que o suportassem e lhe conferissem significado, e ainda de duas apresentações do documento em *eXtensible Stylesheet Language* (XSL). A concretização das duas apresentações em XLS visavam produzir *outputs* em HTML e em *látex*, sendo posteriormente fixado que apenas seria necessário desenvolver uma das apresentações.

Tendo em consideração, também que a minha pré-tese ainda se encontra numa fase demasiado embrionária, achei por bem submeter à apreciação, e como suporte do projecto, a estruturação e processamento deste próprio relatório, que para além de ser entregue em formato PDF poderá ser gerado em formato HTML, como resultado prático do projecto levado a cabo.

1.3. Motivação e Objectivos

A metalinguagem XML, suportada por DTD's ou *Schemas* (ou não) concentra-se na estrutura da informação e não na sua aparência. De forma a visualizar documentos XML requer-se a sua formatação, o que na prática significa que um mesmo documento XML poderá originar múltiplos formatos. Na prática, isto frequentemente significa converter o documento XML em HTML, o que pode ser realizado através de uma linguagem de transformação, sendo aqui usada XSL, tal como solicitado.

A linguagem XSL é usada para especificar a transformação de documentos XML, partindo de um documento XML e transformando-o noutro documento XML. A conversão para HTML consiste simplesmente num caso especial da transformação XML.

De modo a implementar os diversos documentos, recorreu-se ao pacote XML Spy (versão 4.3). Este constitui uma solução para o desenvolvimento de aplicações baseadas em XML, possibilitando uma produção e manutenção simplificadas de documentos XML, *Schemas* XML, bem como *Stylesheets* XSLT, cobrindo as principais vertentes do XML, designadamente: edição e validação XML, edição e validação de DTD's e *Schemas*, edição e transformação XSL.

Pretendia-se neste projecto atacar cada uma das vertentes do XML estudadas ao longo das sessões, coincidindo exactamente com aquelas oferecidas pelo pacote XML Spy, que alias serviu como base de trabalho ao longo das aulas práticas.

O desenvolvimento do projecto no âmbito da disciplina de Processamento Estruturado de Documentos visando a sua avaliação subleva um conjunto de objectivos respeitantes à aplicação prática do corpo de conhecimentos ministrados ao longo do curso, designadamente: estruturação de documentos em XML, produção de DTD's e *Schemas* para validação de documentos, construção e aplicação de transformações XSL a documentos XML.

1.4. Estrutura do Relatório

As subseqüentes secções deste relatório encontram-se organizadas do modo seguinte: o segundo capítulo cobre o documento XML e a respectiva estrutura abordando especificamente em duas subsecções o DTD e o *Schema*; o terceiro capítulo versa a transformação XSL do documento XML possibilitando a visualização deste relatório em HTML; o último capítulo expõe as conclusões emanadas do projecto, regista o nível de cumprimento dos objectivos traçados para o projecto e aponta algumas orientações relativas a desenvolvimentos ulteriores.

Fazem parte integrante deste relatório, ainda, os anexos 1, 2 e 3, onde são detalhadas as codificações respeitantes, respectivamente, ao DTD, ao *Schema* e ao XSL.

2. O documento XML

Uma rápida análise a este relatório transmite uma ideia muito concreta da estruturação que lhe está subjacente. É precisamente a sua organização que será aqui considerada efectuando-se um levantamento detalhado que possibilite a extracção da sua estrutura e a consequente representação na ferramenta empregue.

Procurarei não assentar numa configuração demasiado vertical de modo a ser viável o aproveitamento do DTD/*Schema* para relatórios genericamente considerados, embora não pretendendo ser demasiado ambicioso, criando, eventualmente, demasiados detalhes, porventura de interesse reduzido.

Uma observação de alto-nível deste relatório, permite identificar que este é constituído por: alguns elementos prévios ao relatório (a que chamarei abertura), uma dedicatória, agradecimentos, um resumo, o corpo do relatório, a bibliografia, e anexos. Considerarei que alguns dos elementos estruturais anteriores poderão ou não ser obrigatórios. Assim são facultativas as partes seguintes: dedicatória, agradecimentos, bibliografia e anexos. As restantes são, naturalmente, obrigatórias.

É assim possível o seguinte pseudo-XML, onde os parêntesis rectos evidenciam componentes opcionais e as reticências o conteúdo dos elementos:

```
<relatorio>
  <abertura> . . . </abertura>
  [<dedicatoria> . . . </dedicatoria>]
  [<agradecimentos> . . . </agradecimentos>]
  <resumo> . . . </resumo>
  <corpo> . . . </corpo>
  [<bibliografia> . . . </bibliografia>]
  [<anexos> . . . </anexos>]
</relatorio>
```

Na componente de abertura, podem-se destacar: a entidade no seio da qual se desenvolve o projecto, a disciplina no qual se enquadra, o título do relatório, a autoria e a data do mesmo, donde:

```
<abertura>
  <entidade> . . . </entidade>
  <titulorel> . . . </titulorel>
  <autor> . . . </autor>
  <data> . . . </data>
</abertura>
```

Na componente entidade considere os atributos escola e departamento, enquanto que na componente disciplina recorri ao atributo ano lectivo.

O resumo do relatório decompõem-se no resumo propriamente dito, composto por um ou mais parágrafos de texto, a área de aplicação do projecto e as palavras-chave do relatório. Temos assim, onde (...) + denota um componente repetitivo obrigatório:

```
<resumo>
  (<paragrafo> . . . </paragrafo>)+
  <areaaplicacao> . . . </areaaplicacao>
  <palavraschave> . . . </palavraschave>
</resumo>
```

O corpo do relatório apresenta-se como uma repetição de várias secções (os capítulos e sub capítulos), sendo obrigatório especificar pelo menos uma secção, ou seja:

```
<corpo>
  (<seccao> . . . </seccao>)+
</corpo>
```

Em cada secção encontra-se um título, seguido de um número opcional de parágrafos e/ou de um número opcional de subsecções, pelo que temos o seguinte pseudo-XML, onde (...) * denota um componente repetitivo facultativo:

```
<seccao>
  <titulo> . . . </titulo>
  (<paragrafo> . . . </paragrafo>)*
  (<seccao> . . . </seccao>)*
</seccao>
```

Constate-se a recursividade da especificação anterior que corresponde na prática à existência secções dentro de outras secções. Evidentemente isto possibilita um aninhamento de secções bem mais profundo do que patenteado no relatório em causa (este).

A bibliografia possui uma estruturação similar à usada para o resumo, se bem que mais restrita:

```
<bibliografia>
  (<paragrafo> . . . </paragrafo>)+
</bibliografia>
```

Os anexos constituem uma parte facultativa que a existir será composta de pelo menos um anexo, isto é:

```
<anexos>
  (<anexo> . . . </anexo>)+
</anexos>
```

Em cada anexo é necessário é obrigatória a especificação de um título e de um corpo:

```
<anexo>
  <título> . . . </título>
  <corpoanexo> . . . </corpoanexo>
</anexo>
```

O corpo do anexo constrói-se sobre um ou mais dos elementos já descritos e por qualquer ordem: parágrafo e/ou secção. O símbolo | denota alternativa, e assim temos:

```
<corpoanexo>
  (<paragrafo> . . . </paragrafo> | <seccao> . . . </seccao>)+
</corpoanexo>
```

Para cada um dos restantes elementos terminais não especificados o seu conteúdo é texto livre.

Nas duas secções seguintes são consideradas as descrições estruturais que suportam e validam o documento XML que estrutura o presente relatório.

2.1. A estrutura em DTD

Nesta secção são apresentados os pormenores subjacentes ao DTD criado para o documento XML, cujo detalhe se transcreve no anexo 1.

```
<!ELEMENT relatorio (abertura, dedicatoria?, agradecimentos?, resumo, corpo, bibliografia?, anexos?)>
```

O elemento de topo (relatório) possui a seguinte estrutura:

```
<!ELEMENT abertura (entidade, disciplina, titulorel, autor, data)>
```

A abertura obedece à construção:

Dada a incorporação de um atributo obrigatório e textual na disciplina (referente ao ano lectivo), a sua descrição fica:

```
<!ELEMENT disciplina (#PCDATA)>
<!ATTLIST disciplina anolectivo CDATA #REQUIRED>
```

O elemento resumo é composto por:

```
<!ELEMENT resumo (paragrafo+, areaaplicacao, palavraschave)>
```

Um pouco mais complexa é a estrutura do corpo do relatório, sendo descrita por:

```
<!ELEMENT corpo (seccao)+>  
<!ELEMENT seccao (titulo, paragrafo*, seccao*)>  
<!ATTLIST seccao ident ID #IMPLIED>
```

O atributo identificador ident visa possibilitar a existência de um identificador único associado a cada secção de forma a ser possível a sua referência na construção de hiperligações dentro do documento (por exemplo na elaboração do índice).

O elemento anexos será provavelmente o elemento mais complexo dentre os usados neste projecto, sendo especificado por:

```
<!ELEMENT anexos (anexo)+>  
<!ELEMENT anexo (titulo, corpoanexo)>  
<!ELEMENT corpoanexo (paragrafo | seccao)+>
```

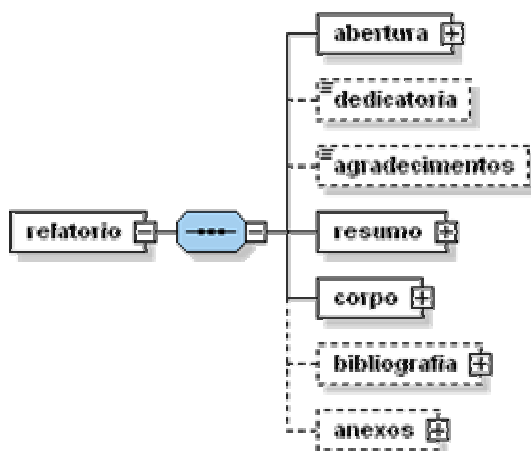
Evidencia-se a alternativa possibilitada nos corpos dos anexos, facultando a criação de anexos com ou sem secções.

Cada um dos restantes elementos aqui expostos mas não descritos são detalhados no anexo 1, sendo a sua compreensão trivial, e como tal de explicação desnecessária.

2.2. O *Schema* do documento

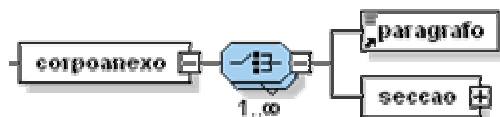
Como alternativa estrutural de validação do documento XML associado a este relatório, os conceitos básicos subjacentes são similares, embora considerados agora no âmbito de um *Schema* e não no de um DTD. O seu detalhe encontra-se no anexo 2, sendo aqui descritos alguns pormenores.

Das descrições anteriores, resulta a existência das sequências seguintes: relatório, abertura, resumo, corpo (apenas com secções), secção, bibliografia, os anexos (com um anexo ou vários) e cada anexo individualmente considerado. Exemplifica-se o caso do relatório:

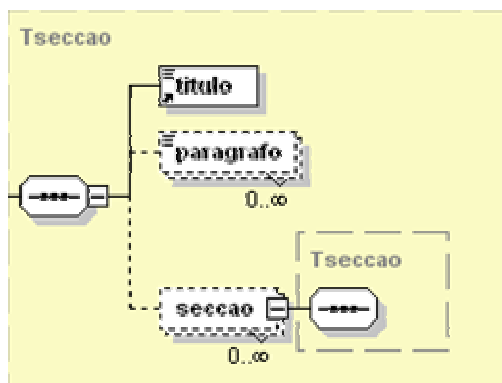


Neste diagrama está patente o carácter opcional (a tracejado) de alguns dos elementos.

Por sua vez o corpo do anexo envolve uma ou mais seleções entre parágrafos e secções. O diagrama referente ao elemento relatório consta da figura seguinte:



Foi constituído apenas um tipo complexo Tseccao, que envolve uma sequência de título, zero ou mais parágrafos e zero ou mais secções e que se expõe na figura seguinte. Naturalmente que uma secção é sempre deste tipo, quer quando associada ao corpo do relatório, quer quando especificada no âmbito do corpo dos anexos ao relatório.



Sugere-se a consulta do anexo 2 para uma completa observação do diagrama referente ao *schema*.

3. A transformação XML2HTML

A transformação implementada em XSL, pretende criar uma página HTML do actual relatório, com uma apresentação similar a este.

Procuraremos acompanhar o processador XSL no tocante aos primeiros modelos (*templates*), após o que consideraremos alguns dos mais relevantes, uma vez que uma consulta detalhada se pode encontrar no anexo 3.

Após carregar os ficheiros XSL e XML (documento fonte), o processador posiciona-se na raiz do documento fonte. Procura então um modelo correspondente à raiz, encontrando:

```
<xsl:template match="/">
  <HTML>
    <HEAD>
      <TITLE>XML2HTML</TITLE>
    </HEAD>
    <BODY>
      <xsl:apply-templates/>
    </BODY>
  </HTML>
</xsl:template>
```

Dado que a raiz se encontra acima do elemento do nível mais superior, torna-se ideal para criar o elemento de topo da árvore resultante. No caso do HTML, isto significa criar as *tags* HEAD e BODY.

Quando encontra *xsl:apply-templates*, o processador desloca-se para o primeiro filho do nó actual. O primeiro filho do nó raiz constitui o elemento de topo do relatório, ou seja, o elemento relatório.

Dada a inexistência de um modelo para o elemento relatório, é despoletado um modelo implícito (predefinido pelo processador):

```
<xsl:template match="* | /">
  <xsl:apply-templates/>
</xsl:template>
```

O modelo predefinido força o processador a carregar o primeiro filho de relatório, que é, abertura. O correspondente modelo é:

```
<xsl:template match="abertura">
  
  <xsl:value-of select="entidade"/> <br/>
  <xsl:value-of select="entidade/@escola"/> <br/>
  <xsl:value-of select="entidade/@departamento"/>
  <br/><br/>
  <h3><xsl:value-of select="disciplina"/> </h3>
  Ano Lectivo de <xsl:value-of select="disciplina/@anolectivo"/>
  <br/><br/><br/><br/>
  <h1><xsl:value-of select="titulorel"/> </h1>
  <h3><xsl:value-of select="autor"/> </h3>
  <xsl:value-of select="data"/>
  <br/>
  <xsl:apply-templates/>
</xsl:template>
```

Note que o modelo corresponde a um *path* relativo dado que o nó actual é relatório. Nele se carrega a imagem que encerra os logótipos da instituição para o documento HTML, assim como o conteúdo do elemento entidade e os valores dos seus atributos (um em cada linha). Realiza-se o mesmo para os elementos disciplina (e seu atributo) título do relatório, autor e data. Finalmente *xsl:apply-templates* faz com que processo continue a partir do elemento data. O filho deste nó é do tipo *text()*, pelo que é despoletado o modelo seguinte que impede o descarregamento de texto para a árvore resultante.

```
<xsl:template match="text() | @*" priority="-1"/>
```

Dado que o texto do elemento data não possui qualquer filho, o processador não pode continuar a progredir em profundidade pelo processa o elemento dedicatória (irmão de abertura), para o qual possui um modelo.

O procedimento descrito continua pelos diversos nós da árvore efectuando uma travessia em profundidade (*deep-first*).

A construção do índice remissivo requer um comentário específico. Assim aquela é iniciada pelo modelo associado ao nó do elemento corpo:

```
<xsl:template match="corpo">
  <h2> <a name="indice"> Índice Remissivo </a> </h2>
  <xsl:apply-templates mode="indice" select="/relatorio/corpo/seccao"/>
  <xsl:apply-templates mode="indice" select="/relatorio/bibliografia"/>
  <p/>
  <xsl:apply-templates mode="indice" select="/relatorio/anexos/anexo"/>
  <xsl:apply-templates select="/relatorio/corpo/seccao"/>
</xsl:template>
```

Neste podemos constatar a geração do título do índice com uma *bookmark* associada, destinado a regressar a este ponto a partir de qualquer outro local na página HTML. É então despoletada uma travessia, despoletada por *xsl:apply-templates mode="indice"*

`select="/relatorio/corpo/seccao"` para recolher os títulos das diferentes secções. Tal travessia é concretizada pelos modelos:

```
<xsl:template match="seccao" mode="indice">
  <a href="{generate-id()}"><xsl:number level="multiple"/>.
  <xsl:value-of select="titulo"/></a> <br/>
  <xsl:apply-templates mode="indice"/>
</xsl:template>

<xsl:template match="text() | @*" mode="indice" priority="-1"/>
```

O segundo modelo evita a apresentação dos conteúdos indesejados no âmbito do índice. No primeiro modelo, podemos constatar a extracção do conteúdo do elemento título a que é associada uma referência para ligação à secção à qual o título respeita.

Para completar a interacção do índice com as secções referenciadas, é considerado o modelo seguinte:

```
<xsl:template match="seccao">
  <xsl:if test="name(parent::node()) = 'corpo'"> <hr/> </xsl:if>
  <h2>
    <a name="{generate-id()}"> <xsl:number level="multiple"/>. </a>
    <xsl:value-of select="titulo"/> </h2>
    <a href="#indice"> [Índice]</a> <p/>
  <xsl:apply-templates/>
</xsl:template>
```

Dado que cada secção possui obrigatoriamente um título, este constitui o elemento capturado pelo modelo, extraindo-se o conteúdo do elemento título, associando-se-lhe uma *bookmark*. De modo a disponibilizar uma forma de retorno ao índice, é também construída uma hiperligação que fica associada ao texto [Índice]. A primeira instrução visa criar uma linha separadora horizontal na página HTML resultante apenas no caso de estarmos perante o início de secções de topo (capítulos). Para tal verifica-se se o pai do nó actual corresponde ao elemento corpo.

Voltando ao modelo do elemento corpo, consta-se uma estratégia similar para o tratamento das entradas no índice referentes à bibliografia, anexos e suas eventuais secções.

Uma última palavra para o modelo de tratamento dos parágrafos:

```
<xsl:template match="paragrafo">
  <p><xsl:value-of select="."/> </p>
  <xsl:if test="@imagem">  </xsl:if>
  <xsl:apply-templates/>
</xsl:template>
```

De modo a colocar imagens na página HTML resultante, o elemento parágrafo foi desenhado de forma a suportar um atributo para efeito, denominado imagem. Naturalmente, caso tal não seja preenchido (dado que não é obrigatório), o parágrafo é de texto livre; caso seja especificado, para além do texto é seguido do carregamento da imagem cujo *path* está associado ao atributo imagem. Neste sentido, a segunda instrução, verifica a especificação da imagem, concretizando a sua saída na página HTML gerada.

4. Conclusões e trabalho futuro

O projecto relatado no presente documento, constitui uma interessante concretização e consolidação do corpo de conhecimentos transmitido no decurso da disciplina de Processamento Estruturado de Documentos do Mestrado/Curso de Especialização em Informática do ano lectivo 2001/2002.

De facto, e tendo em consideração os objectivos traçados no primeiro capítulo deste relatório, verifica-se um cumprimento integral dos mesmos, designadamente: estruturação de documentos em XML, produção de DTD's e Schemas para validação de documentos, construção e aplicação de transformações XSL a documentos XML.

Naturalmente que algum desenvolvimento futuro pode ser considerado dado que podem ser levadas em linha de conta a existência de anotações para: ênfatização de texto, tratamento de tabelas, etc. Contudo, no âmbito do projecto em causa, estas características adicionais não me parecem comportar uma mais valia assinalável dado que pouco conhecimento acrescenta, para além do demonstrado. Não ponho porém em causa a sua utilidade e relevância prática e que podem ser encarados no âmbito de um desenvolvimento subsequente.

Um potencialmente e interessante acréscimo passa pela criação de um índice de figuras, para o qual o elemento parágrafo já se encontra preparado, passando apenas pelo seu tratamento a nível do XSL, em que o conteúdo do parágrafo pode ser tratado de duas formas independentes: uma para texto e outra para imagem. Neste último caso, o conteúdo do elemento constituiria a legenda da figura, requerendo assim mais uma travessia da árvore documental abstracta.

Bibliografia

Elliote Rusty Harold. XML bible (2nd Edition). IDG Books, 2001.

Elliote Rusty Harold and W. Scott Means. XML in a Nutshell. O'Reilly, 2001.

Elliote Rusty Harold. Chapter 17 of the XML Bible (2nd Edition). Disponível *on-line* em <http://www.ibiblio.org/xml/books/bible2/chapters/ch17.html>

Elliote Rusty Harold. Chapter 24 of the XML Bible (2nd Edition). Disponível *on-line* em <http://www.ibiblio.org/xml/books/bible2/chapters/ch24.html>

José Carlos Ramalho. Anotação Estrutural de Documentos e sua Semântica. Universidade do Minho, 2000.

W3C. Diversos *on-line* em <http://www.w3c.org>

ANEXOS

Anexo 1

Estrutura DTD

Apresenta-se o DTD completo do documento XML referente ao presente relatório.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT relatorio (abertura, dedicatoria?, agradecimentos?, resumo,
corpo, bibliografia?, anexos?)>
<!ELEMENT abertura (entidade, disciplina, titulorel, autor, data)>
<!ELEMENT entidade (#PCDATA)>
<!ATTLIST entidade
    escola CDATA #REQUIRED
    departamento CDATA #REQUIRED
>
<!ELEMENT disciplina (#PCDATA)>
<!ATTLIST disciplina
    anolectivo CDATA #REQUIRED
>
<!ELEMENT titulorel (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT data (#PCDATA)>
<!ELEMENT dedicatoria (#PCDATA)>
<!ELEMENT agradecimentos (#PCDATA)>
<!ELEMENT resumo (paragrafo+, areaaplicacao, palavraschave)>
<!ELEMENT areaaplicacao (#PCDATA)>
<!ELEMENT palavraschave (#PCDATA)>
<!ELEMENT corpo (seccao)+>
<!ELEMENT bibliografia (paragrafo+)>
<!ELEMENT anexos (anexo)+>
<!ELEMENT anexo (titulo, corpoanexo)>
<!ELEMENT corpoanexo (paragrafo | seccao)+>
<!ELEMENT seccao (titulo, paragrafo*, seccao*)>
<!ATTLIST seccao
    ident ID #IMPLIED
>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT paragrafo (#PCDATA)>
```

Anexo 2

Estrutura Schema

Apresenta-se o *Schema* completo do documento XML referente ao presente relatório.

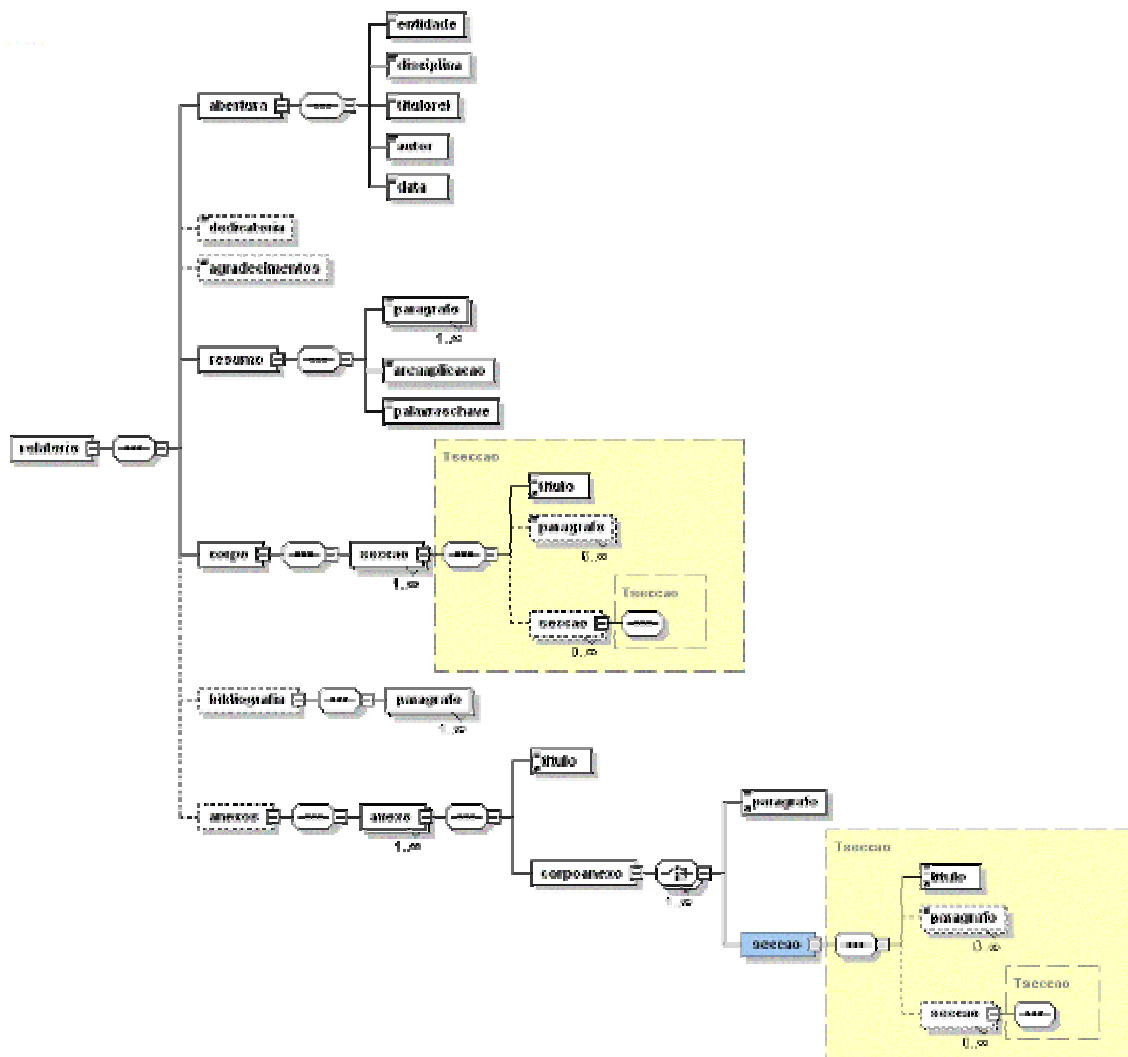
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- edited with XML Spy v4.3 U (http://www.xmlspy.com) by CC (UM) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="paragrafo">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="ident" type="xs:ID" use="optional"/>
          <xs:attribute name="imagem" type="xs:string" use="optional"/>
          <xs:attribute name="legenda" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="relatorio">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="abertura">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="entidade">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute name="escola" type="xs:string" use="required"/>
                      <xs:attribute name="departamento" type="xs:string" use="required"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="disciplina">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute name="anolectivo" type="xs:string" use="required"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="titulorel" type="xs:string"/>
              <xs:element name="autor" type="xs:string"/>
              <xs:element name="data" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="dedicatoria" type="xs:string" minOccurs="0"/>
        <xs:element name="agradecimentos" type="xs:string" minOccurs="0"/>
        <xs:element name="resumo">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="paragrafo" type="xs:string" maxOccurs="unbounded"/>
              <xs:element name="areaaplicacao" type="xs:string"/>
              <xs:element name="palavraschave" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="corpo">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="seccao" type="Tseccao" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="bibliografia" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="paragrafo" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="anexos" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="anexo" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="titulo" />
            <xs:element name="corpoanexo">
              <xs:complexType>
                <xs:choice maxOccurs="unbounded">
                  <xs:element ref="paragrafo" />
                  <xs:element name="seccao" type="Tseccao" />
                </xs:choice>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="Tseccao">
  <xs:sequence>
    <xs:element ref="titulo" />
    <xs:element name="paragrafo" type="xs:string" minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="seccao" type="Tseccao" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:element name="titulo" type="xs:string" />
</xs:schema>

```

O diagrama completo do *Schema* correspondente ao código anterior apresenta-se de seguida.



Anexo 3

Transformação XSL

Expõe-se a transformação XSL que converte o documento XML em HTML.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:output indent="yes" method="html" encoding="iso-8859-1" omit-xml-
declaration="yes"/>

  <xsl:template match="/">
    <HTML>
      <HEAD>
        <TITLE>XML2HTML</TITLE>
      </HEAD>
      <BODY>
        <xsl:apply-templates/>
      </BODY>
    </HTML>
  </xsl:template>

  <xsl:template match="abertura">
    
    <xsl:value-of select="entidade"/> <br/>
    <xsl:value-of select="entidade/@escola"/> <br/>
    <xsl:value-of select="entidade/@departamento"/>
    <br/><br/>
    <h3><xsl:value-of select="disciplina"/> </h3>
    Ano Lectivo de <xsl:value-of select="disciplina/@anolectivo"/>
    <br/><br/><br/><br/>
    <h1><xsl:value-of select="titulorel"/> </h1>
    <h3><xsl:value-of select="autor"/> </h3>
    <xsl:value-of select="data"/>
    <hr/>
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="dedicatoria">
    <h2> Dedicatória </h2>
    <table width="250"> <tr> <td>
    <p> <xsl:value-of select="."/> </p>
    </td> </tr> </table>
    <hr/>
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="agradecimentos">
    <h2> Agradecimentos </h2>
    <table width="250"> <tr> <td>
    <p> <xsl:value-of select="."/> </p>
    </td> </tr> </table>
    <hr/>
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="resumo">
    <h2> Resumo </h2>
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="paragrafo">
```

```

    <p><xsl:value-of select="."/> </p>
    <xsl:if test="@imagem" >  </xsl:if>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="resumo/areaaplicacao">
  <p> Área de aplicação: <xsl:value-of select="."/> </p>
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="resumo/palavraschave">
  <p> Palavras-Chave: <xsl:value-of select="."/> </p>
  <hr/>
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="corpo">
  <h2> <a name="indice"> Índice Remissivo </a> </h2>
  <xsl:apply-templates mode="indice" select="/relatorio/corpo/seccao"/>
  <xsl:apply-templates mode="indice" select="/relatorio/bibliografia"/>
  <p/>
  <xsl:apply-templates mode="indice" select="/relatorio/anexos/anexo"/>
  <xsl:apply-templates select="/relatorio/corpo/seccao"/>
</xsl:template>

<xsl:template match="seccao">
  <xsl:if test="name(parent::node()) = 'corpo'"> <hr/> </xsl:if>
  <h2>
  <a name="{generate-id()}"> <xsl:number level="multiple"/>. </a>
  <xsl:value-of select="titulo"/> </h2>
  <a href="#indice"> [Índice]</a> <p/>
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="bibliografia">
  <hr/>
  <h2>
  <a name="{generate-id()}"> />
  Bibliografia </h2>
  <a href="#indice"> [Índice]</a> <p/>
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="anexos">
  <hr/>
  <h2> Anexos </h2>
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="anexo">
  <h2>
  Anexo <a name="{generate-id()}"> <xsl:number/> - </a>
  <xsl:value-of select="titulo"/> </h2>
  <a href="#indice"> [Índice]</a> <p/>
  <xsl:apply-templates/>
  <hr/>
</xsl:template>

<xsl:template match="text() | @*" priority="-1"/>
<!-- Geração dos Índices:##### -->
<xsl:template match="seccao" mode="indice">
  <a href="{generate-id()}"><xsl:number level="multiple"/>.
  <xsl:value-of select="titulo"/></a> <br/>
  <xsl:apply-templates mode="indice"/>
</xsl:template>

<xsl:template match="bibliografia" mode="indice">
  <a href="{generate-id()}">
  Bibliografia </a> <br/>
  <xsl:apply-templates mode="indice"/>
</xsl:template>

<xsl:template match="anexo" mode="indice">
  <a href="{generate-id()}"> Anexo <xsl:number level="single"/> -
  <xsl:value-of select="titulo"/></a> <br/>
  <xsl:apply-templates mode="indice"/>

```



```
</xsl:template>
<xsl:template match="text() | @" mode="indice" priority="-1"/>
<!-- ##### -->
</xsl:stylesheet>
```