

## Linguagens de Anotação XML, SGML, ...

José Carlos L. Ramalho  
 jcr@di.uminho.pt  
 gEPL - DI - UM



## Conteúdos

- Anotação
  - O que é? Anotar para quê?
- SGML - a história e o passado
  - ciclo de vida dos documentos
- A Evolução forçada pela Web
  - HTML e SGML: Web SGML
- XML
  - Conceitos
  - Ciclo de vida e Aplicações

## O que é um documento?



- uma imagem
- uma sequência de palavras
- informação estruturada
- um ser vivo (não no nosso caso)
  - ♦ Dublin Core - "Metadata Workshop"

## O que é que fazemos com eles?



- Eu quero imprimir o meu relatório
- O meu chefe quer pô-lo na Internet (WWW)
- O meu colega quer refazer alguns dos cálculos na folha de cálculo
- Todos querem que fique seguramente arquivado



**Começa a surgir um problema ...**

## Que "suportes" utilizamos?



- Eu utilizo o LaTeX na minha NeXT
- O meu colega usa o MSWinWord no seu PC
- O nosso chefe ainda usa o WordPerfect no DOS
- As nossas secretárias utilizam o Word no Macintosh
- Ainda há alguém no departamento a usar o troff

**Temos um problema!!!**



## O Problema:

### O que é que necessitamos?

- Um suporte neutro para a nossa informação
- Independente de plataformas SW e HW

### O que é que o mercado oferece?

**XML**

eXtended Markup Language

## Motivação

- Maioria da informação em suporte digital: tradicionalmente em MSWord, Wordperfect, Access, Excel, Acrobat.
- Transferência e **manipulação** de documentos entre plataformas dispendiosa
- **Manutenção** difícil implica uma baixa longevidade
- **Normalização** muito difícil
- Custos de distribuição podem ser altos
- A migração para a nova tecnologia que se vai apresentar **pode não ser uma opção!**...

## XML: O que é?

- Uma linguagem para definir linguagens específicas de anotação: SMDL, HTML, EAD, DocBook, TEI, ChessML.
- É independente do suporte (HW e SW).
  - Implica a separação do conteúdo e da forma.
- É um standard ISO
  - ideal para armazenamento e intercâmbio

## O que é a Anotação dum texto?

Um meio de tornar explícita uma interpretação desse texto.

### Exemplo1: sinais de pontuação

Olá. Tudo bem.  
Olá! Tudo bem?

### Exemplo2: anotações

"O rei <nome>Afonso Henriques</nome> queria ..."

## Anotar um documento

- A tarefa de anotar um documento num sistema de processamento compreende as seguintes fases:
  - analisar a estrutura da informação e os atributos que a caracterizam
  - determinar quais as funções de processamento que produzirão o formato desejado para cada elemento
  - inserir as etiquetas/anotações no texto

## Pausa para pensar

- Exercício: anotar o poema
- Exercício: anotar o texto dum sumário
- Exercício: anotar página de livro
- ...

## O Poema

"Soneto Já Antigo"  
(Álvares de Campos)

Olha, Daisy: quando eu morrer tu hás-de dizer aos meus amigos ai de Londres, embora não o sintas, que tu escondes a grande dor da minha morte. Irás de

Londres p'ra lorque, onde nasceste (dizes que eu nada que tu digas acredito), contar àquele pobre rapazito que me deu horas tão felizes,

embora não o saibas, que morri... Mesmo ele, a quem eu tanto julguei amar, nada se importará... Depois vai dar

a notícia a essa estranha Cecily que acreditava que eu seria grande... Raios partam a vida e quem lá ande!

(1922)

## Poema: elementos estruturais

"Soneto Já Antigo"  
(Álvaro de Campos)

Olha, Daisy: quando eu morrer tu hás-de dizer aos meus amigos aí de Londres, embora não o sintas, que tu escondes a grande dor da minha morte. Irás de

Londres p'ra Lorque, onde nasceste (dizes que eu nada que tu digas acreditado), contar àquele pobre rapazito que me deu horas tão felizes,

em bora não o saibas, que morri... Mesmo ele, a quem eu tanto julguei amar, nada se importará... Depois vai dar

a notícia a essa estranha Cecily que acreditava que eu seria grande... Raios partam a vida e quem lá ande!

(1922)

**Poema:** título, autor, corpo, data

**Corpo:** quadra, quadra, terno, terno.

**Quadra:** verso, verso, verso, verso

**Terno:** verso, verso, verso

**Verso:** (texto | nome)+

**Nome:** texto

## Os Sumários

Processamento Estruturado de Documentos

José Carlos Ramalho  
jer@di.uminho.pt  
http://www.di.uminho.pt/~jer

[2000.10.02]  
Anotação de Documentos: um pouco de história.  
Linguagens de Anotação como meta-linguagens: o SGML e o XML.  
Anotação Descritiva. Ciclo de vida dos documentos estruturados.  
...

[2000.10.18]  
Resolução da ficha teórico prática número 2 no laboratório.

## Tipos de Anotação

**Procedimental** - define qual o processamento a ser realizado em determinados pontos do documento.

```
Caros senhores,
.esp_vert 4
Escrevo-vos a ... items:
.indent 4
1. Qual a qualificação necessária
.indent 4
2. Datas
...
.esp_vert 4
Obrigado
```

## Tipos de Anotação 2 (cont.)

**Descritiva** - utiliza etiquetas para apenas classificar as componentes do documento.

```
<abre>Caros senhores.</abre>
<corpo>
Escrevo-vos a ... items:
<listan>
<item> Qual a qualificação necessária
<item> Datas
</listan>
...
</corpo>
<fecha>Obrigado</fecha>
```

## Funções da Anotação

- representar todos os caracteres de um texto
- identificar a estrutura do texto
- reduzir o texto a uma ordem linear (árvore)
- representar informação contextual
- distinguir o que é texto do que é anotação

## Linguagem de Anotação

- Especifica como distinguir a anotação do texto
- Especifica **que** anotações são **necessárias** e **quais** são **permitidas**
- Especifica **onde** as anotações são **necessárias** e **onde** são **permitidas**
- Define o significado da anotação

**O XML tem estas características todas ... excepto a última**

## Documentos

- Tradicionalmente: um bloco indivisível
- Nesta abordagem: uma estrutura

## Documentação Estruturada

- Vantagens (c/ anotação descritiva)
  - Portabilidade e intercâmbio entre plataformas
  - reutilização
  - armazenamento e consulta
  - baixos custos de distribuição
  - manutenção e acesso
  - longevidade
  - reconfiguração do aspecto visual

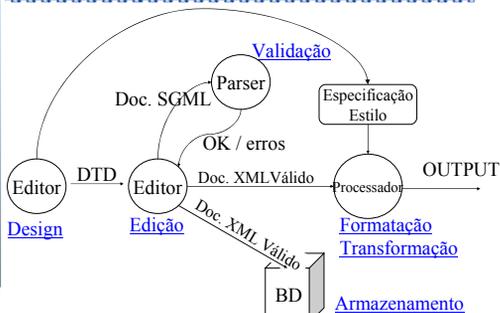
## Documentação Estruturada

- Desvantagens
  - grande investimento inicial
    - capital
    - tempo
  - o XML veio aligeirar estes parâmetros...

## XML: premissa heurística

- Os documentos têm estrutura que se quer preservar e manipular:
  - na sua criação
  - no seu processamento
  - na sua interrogação (=> armazenamento)
    - Select EXCERTO with NOME="Ana\*"
    - Select TITULO with parent EXCERTO

## XML: Ciclo de vida



## XML: conceitos

- \* **element** - definição de tipo de um objecto, componente estrutural do documento
- \* **attribute** - propriedade de um elemento
- \* **entity** - "stream" de dados
- \* **notation** - notação/formato externo
- \* **document type** - conjunto de objectos e entidades, semelhante a uma gramática ou esquema de base de dados

## Os documentos têm estrutura!

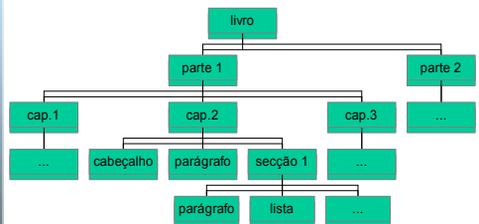
Uma *definição* contém um *termo técnico* seguido dum *texto descritivo*



```
<!ELEMENT definicao (termo, descricao)>
```



## O texto forma uma hierarquia ordenada



## Indicadores de estrutura

### no texto...

- espaço, fontes diferentes, gráficos...
- derivados da semântica: nomes, datas, lugares, ...

### em XML...

- delimitadores explícitos
- inferida do contexto



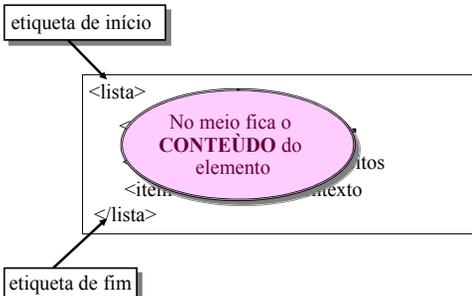
## XML: exemplo

```
<lista>
  <cabec>Os delimitadores podem ser: </cabec>
  <item> explícitos </item>
  <item> inferidos do contexto </item>
</lista>
```

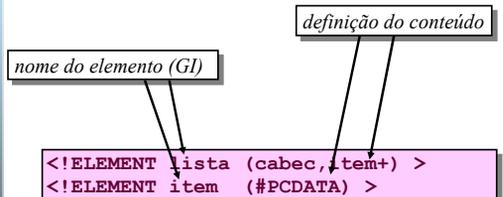
*O elemento do tipo LISTA é formado por um elemento CABEC, seguido por dois elementos do tipo ITEM.*



## Ocorrência de um elemento



## Definição de um elemento



## Definição do CONTEÚDO

- outros elementos especificados
- ANY (qq elemento especificado)
- EMPTY (nada, vazio)
- #PCDATA (texto)
- uma mistura de elementos com #PCDATA

## Expressão de Conteúdo: sintaxe

- sequência
  - a,b      a seguido de b
  - a|b      a ou b mas não ambos
- ocorrência
  - a          um e apenas um
  - a?        opcionalmente um (0 ou 1)
  - a\*        zero ou mais
  - a+        um ou mais

## Exemplo: o poema

Poema: título, autor, corpo, data  
 Corpo: quadra, quadra, terno, terno.  
 Quadra: verso, verso, verso, verso  
 Terno: verso, verso, verso  
 Verso: (texto | nome)+

```
<!ELEMENT poema (título,autor,corpo,data) >
<!ELEMENT corpo (quadra,quadra,terno,terno) >
<!ELEMENT quadra (verso,verso,verso,verso) >
<!ELEMENT terno (verso,verso,verso) >
<!ELEMENT verso (#PCDATA |nome)* >
```

## Exercício (pausa)

Especificar o DTD para os Sumários

## Um elemento pode ter atributos

- para conter informação para além do tipo e do contexto
- para identificação de ocorrências específicas de elementos
- para fazer algumas validações (poucas)

nome do atributo

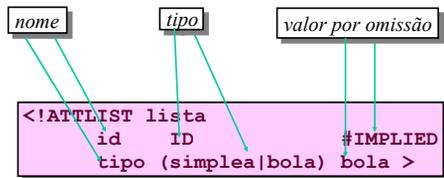
valor do atributo

```
<lista tipo=bola id=L123>
  <item id=L123.1> delimitadores explicitos
  <item id=L123.2> inferidos do contexto
</lista>
```

## Ex: anotação morfo-sintáctica

```
<quadra>
<verso><verbo tempo="imperativo"
pessoa="2s">Olha</verbo>,<nome> Daisy</nome>: quando eu morrer tu
hás-de</verso>
<verso><verbo tempo="infinitivo">dizer</verbo> aos meus amigos ai
de <nome>Londres</nome>,</verso>
<verso>embora não o sintas, que tu escondes</verso>
<verso>a grande dor da minha morte. Irás de</verso>
</quadra>
```

## Definição de um atributo



os identificadores de nomes e tipos devem ser únicos dentro dum elemento

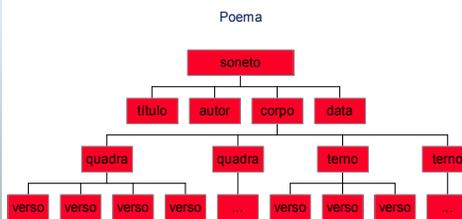
## Tipos de atributo

- **ID** um identificador único dentro o documento actual
- **IDREF** referência a um identificador definido algures no documento corrente
- **CDATA** texto
- **NAME, NUMBER, NMTOKEN**
- **ENTITY** o nome duma entidade definida no documento corrente
- uma enumeração/lista de valores específicos (não pode haver repetições na lista)

## Valores possíveis

- **#REQUIRED** (obrigatório)
- **#FIXED** (constante)
- **#IMPLIED** (opcional)
- *valor explicito*

## A estrutura do nosso poema



## Definição: DTD

```
<!-- declarações para os elementos -->
<!ELEMENT poema -- (titulo, autor, corpo, data) >
<!ELEMENT titulo -- (#PCDATA) >
<!ELEMENT autor -- (#PCDATA) >
<!ELEMENT corpo -- ((quadra|terno)+) >
<!ELEMENT quadra -- (verso, verso, verso, verso) >
<!ELEMENT terno -- (verso, verso, verso) >
<!ELEMENT verso -- (#PCDATA|nome)+ >
<!ELEMENT nome -- (#PCDATA) >
<!ELEMENT data -- (#PCDATA) >
```

Se poema fosse soneto:

```
<!ELEMENT corpo -- (quadra, quadra, terno, terno)>
```

## Poema anotado

```
<poema><titulo>Soneto Já Antigo</titulo>
<autor>Álvoro de Campos</autor>
<quadra>
<verso>Olha,<nome> Daisy</nome>: quando eu morrer tu hás-de</verso>
<verso>dizer aos meus amigos ai de <nome>Londres</nome>,</verso>
<verso>embora não o sintas, que tu escondes</verso>
<verso>a grande dor da minha morte. Irás de</verso>
</quadra><quadra>
<verso><nome>Londres</nome> p'ra <nome>lorque</nome>, onde nasceste
(dizes</verso>
<verso>que eu nada que tu digas acredito,</verso>
<verso>contar àquele pobre rapazito</verso>
<verso>que me deu horas tão felizes,</verso>
</quadra><terno>
<verso>em bora não o saibas, que morri...</verso>
...</terno>
<data>1922</data></poema>
```

## software XML

- *parsers*: *NSGMLS*, *SP*
  - validam um DTD
  - verificam se um documento está de acordo com o DTD
  - devolvem como output o documento em formato esis
- *editores*: *XMetal*, *Emacs*, *Adept*, *Xml Spy*...
  - usam o DTD para controlar a edição do texto
- *tradutores*: *Omnimark*, *Balise*, *Perl*
  - *up/down translation* entre XML e outro formato
- *motores IR*: *InfoManager*, *Altavista*, ...
  - utilizam a estrutura do XML para otimizar as procuras

## Instruções de Processamento

- Indicações para processadores externos

```
<?xml version="1.0" encoding="ISO-8859-1"?)>
```

```
<?Tex \newpage?)>
```

```
<?HTML <HR?)>
```

## Notações

- Apenas se podem declarar
- Permitem suportar dados não-XML

```
<!NOTATION TeX SYSTEM "formatos/texto">
<!NOTATION GIF SYSTEM "formatos/imagem">
```

## Secções Marcadas

- Úteis para incluir exemplos de XML

```
<![CDATA [
The <p> tag is used for paragraphs
]]>
```

- E para processamentos condicionais

```
<ENTITY % RTF "INCLUDE">
<ENTITY % HTML "IGNORE">
```

```
<![ %RTF; [
&my-wmf;
]]>
```

```
<![ %HTML; [
&my-gif;
]]>
```

## Tipos de Entidade

- geral
  - apenas usada no documento
  - a sua expansão está definida no DTD
- externa
  - apenas usada no documento
  - a sua expansão é o identificador dum objecto externo
- paramétrica
  - apenas usada no DTD
  - a expansão pode ser qq coisa

## Entidades Gerais

*Utilização*: abreviaturas, caracteres especiais e símbolos etc.

*Exemplo*:

```
Isto devia ser um curso de
&xml;
dado pelo Jos&eacute;; Carlos
```

*Definição*:

```
<!ENTITY xml "eXtended Markup Language">
<!ENTITY eacute "é" >
<!ENTITY eacute "„" >
<!ENTITY eacute SDATA "[eacute]">
```

## Entidades Gerais (2)

- Também podem incluir texto anotado:

```
<!ENTITY rodape ' <hr size="1"/>
<p>Copyright 2001 by
<a href="http://www.di.uminho.pt/~jcr"><p>
</p>
```

## Entidades externas

*Utilização:* incluir ou referenciar um ou mais objectos externos (ficheiros, imagens)

*Exemplo:*

```
&chap1; &chap1fig;
<!-- preciso escrever chap 2 -->
<p>Eis uma referência para
<xref doc=myHome>a minha home page</xref>
```

*Definição:*

```
<!ENTITY chap1 SYSTEM "chap1.sgm">
<!ENTITY chap1fig SYSTEM "gifs/chap1fig.jpg"
NDATA JPG>
<!ENTITY myHome SYSTEM
"http://orunner.di.uminho.pt/~jcr" NDATA HTML>
```

## Entidades paramétricas

*Utilização:* parametrização de declarações e control de secções marcadas

*Exemplo:*

```
%ISOLat1;
<!ATTLIST exemplo %a.glob>
<![ %HTML [ ...
]]>
```

*Definição:*

```
<!ENTITY % ISOLat1 PUBLIC "ISO 8879-
1986//ENTITIES Added Latin 1//EN">
<!ENTITY % a.glob "ident ID #IMPLIED">
<!ENTITY % HTML "INCLUDE">
```

## Entidades paramétricas (exs)

```
<!ENTITY % para-content "(#PCDATA | imagem
| nome | lugar | ... )*">
<!ELEMENT paragrafo %para-content;>
```

## Juntando as peças

*Um documento XML compreende:*

- uma declaração XML
  - <?xml version="1.0" ... ?>
- um DTD (opcional)
- a instância do documento

## A Declaração do Tipo de Documento

- contém as declarações para
  - elementos, atributos, entidades
- define uma hierarquia única de elementos
- é normalmente invocada por referência
- pode ser modificada por um *subset do DTD*

## Alguns exemplos

```
<!DOCTYPE poema [
<!ELEMENT poema - o (quadra+)>
<!ELEMENT quadra - o (#PCDATA)
]>
```

```
<!DOCTYPE poema PUBLIC
"-//jcr//DTD poema 1.0">
```

```
<!DOCTYPE poema SYSTEM "poema.dtd" [...]>
```

## Declarações múltiplas

*Para elementos*

- não é permitida

*Para entidades*

- é permitido: a primeira é que prevalece
- a declaração tem que preceder o uso

## Processamento Condicional

*no dtd*

```
<!ENTITY % my-gifs "IGNORE">
<!ENTITY % my-wmfs "IGNORE">
<![ %my-gifs [
<!-- imagens para o HTML -->
]]>
<![ %my-wmfs [
<!-- imagens para RTF, PDF, MIF -->
]]>
```

*na parametrização do dtd*

```
<!ENTITY % my-wmfs "INCLUDE">
```

## Até ao SGML ...

- A ideia de que a documentação estruturada podia ser trocada e manipulada se fosse produzida de uma forma standard vem dos anos 60:
  - GCA → GenCode
  - IBM → GML
- Em 1980 é formado o comité ANSI para as linguagens de processamento de texto com o objectivo de encontrar uma norma para a especificação, a definição e a utilização de anotações em documentos.
- Em 1986, é lançado o SGML como o ISO 8859



## SGML (the beginning)

- Apesar de todo o poder expressivo e versatilidade não se conseguiu impôr
- Grande complexidade
- Grande investimento inicial de capital e de tempo

## O HTML é SGML ...

Será que  descende de  ?

No fim dos anos 80 Tim Berners-Lee misturou:

- um conjunto de tags dum DTD utilizado no CERN
- juntou-lhe uma especificação de estilo
- e mais um elemento revolucionário: os links

**O que deu origem ao HTML**

## HTML: exemplo

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
<META NAME="Author" CONTENT="Jose Carlos Ramalho">
<META NAME="GENERATOR" CONTENT="Mozilla/4.01 [en] (Win95; I) [Netscape]">
<TITLE>soneto de &Aacute;lvares de Campos</TITLE>
</HEAD>
<BODY>
<H2>
"Soneto J&aacute;ante; Antigo"</H2>
<H3>
(&Aacute;lvares de Campos)</H3>
<P>Olha, <I><FONT COLOR="#3333FF">Daisy</FONT></I>: quando eu morrer tu
h&aacute; de
<BR>dizer aos meus amigos &aacute; de, de <I><FONT COLOR="#3366FF">Londres</FONT></I>,
<BR>embora n&atilde;o os sintas, que tu escondes
<BR>a grande dor da minha morte. Ir&aacute; de
...
```

## HTML (the beginning)

- A sua grande simplicidade foi o factor determinante para a grande adesão que teve
- A sua grande simplicidade foi também o início de uma época problemática
  - atingiu rapidamente o seu limite
  - começaram a proliferar extensões não normalizadas

## O sentido ecuménico do HTML

- o HTML representa a linguagem de anotação que as pessoas podem tomar como certa(!!!).
- quanto mais universal fôr uma linguagem menos especifica se torna, quanto maior fôr o contexto de aplicação menor será a capacidade de expressão semântica.

## XML (como surgiu?)

O poder do SGML      A simplicidade do HTML

**XML**

Uma linguagem de anotação aberta

## XML (conceitos)



- não há cruzamento de tags      • pertence a uma classe (DTD)
  - <A>olá <B>estás </A> ...
- pode-se inferir um DTD
- torna o pós-processamento mais específico.

## SGML → XML

- Fechar todas as "tags" que se abrirem
- Converter os elementos vazios
  - <HR> => <HR/>
  - def. SGML: <!ELEMENT HR -o EMPTY>
  - def. XML: <!ELEMENT HR EMPTY>
- Colocar os valores dos atributos dentro de aspas
  - <IMG SRC="foto.jpg"/>

## XML: exemplo - dtd do poema

```

<!-- declarações para os elementos -->
<ELEMENT poema (titulo, autor, corpo, data) >
<ELEMENT titulo (#PCDATA) >
<ELEMENT autor (#PCDATA) >
<ELEMENT corpo ((quadra|terno)+) >
<ELEMENT quadra (verso, verso, verso, verso) >
<ELEMENT terno (verso, verso, verso) >
<ELEMENT verso (#PCDATA|nome) + >
<ELEMENT nome (#PCDATA) >
<ELEMENT data (#PCDATA) >

```

## XML: exemplo - instância

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="poema.css"?>
<!DOCTYPE poema SYSTEM "poema.dtd">
<poema>
<titulo>"Soneto J&#225; Antigo</titulo>
<autor>(&#193;lvaro de Campos)</autor>
<corpo>
<quadra>
<verso><nome>Daisy</nome>; quando eu morrer tu h&#225;s de</verso>
<verso>dizer aos meus amigos a&#237; de <nome>Londres</nome>;</verso>
<verso>embora n&#227;o o sintas, que tu escondes</verso>
<verso>a grande dor da minha morte. Ir&#225;s de</verso>
</quadra>
<quadra>
<verso><nome>Londres</nome> p'ra <nome>lorque</nome>, onde nasceste (dizes</verso>
<verso>que eu nada que tu digas acredito);</verso>
<verso>contar &#224;quele pobre rapazito</verso>
<verso>que me deu horas t&#227;o felizes;</verso>
</quadra>
...

```

## XML (áreas de utilização)

- Produção de documentação em geral
- Produção de corpora anotado
- Registos Clínicos: H7
- Arquivos: EAD
- Química: CML
- Xadrez: ChessML
- Música: MusicML
- ...

## Casos reais

- Linguagens definidas em SGML/XML
  - DocBook XML 1.5
  - SMDL- música
  - CML- química
  - SMIL- multimedia e interface
  - MathML- matemática
  - H7- medicina
  - XML ED- comércio electrónico

## Problema interessante

- Normalização de conteúdos:
  - Afonso
  - Aphonso
  - Affonso
  - o Rei

## Conclusão

**Libertem a vossa informação,  
usem Anotação Descritiva!!**

Perguntas?

## Documento como um todo

1760

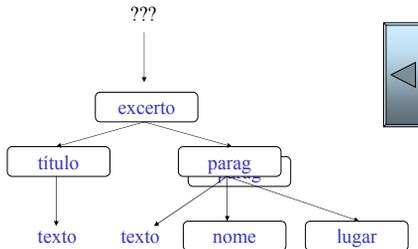
Em 14 de Junho deste anno foi mandado sahir immediatamente da Corte o cardeal Accioalolli e dentro em 4 dias do Reino, para o que se lhe deu decente transporte e acompanhamento. Era Nuncio de sua Santidade o Papa Clemente 13. Aqui principiou a rotura com a Sé Apóstolica. Fes-se publico o motivo pela informação que se mandou a Francisco de Almada residente na Corte de Roma, dando-se por fundamento maior o não pôr luminarias por ocasião do cazamento que se fes no dia seis de Junho do Senhor Infante D. Pedro com sua sobrinha a Excelentissima Senhora D. Maria, princesa do Brasil, o que o dito Cardeal não fizera, porque lho não fizeram a saber por carta de officio, quando os mais embaixadores sem isso o fizeram. O caso he que querião afasta-lo da Corte e tambem elle não seguia a conduta della por aquelles tempos.

Extraído de “Memórias de Inácio Peixoto dos Santos”

## Documento estruturado

```
<EXCERTO><TITULO>1760</TITULO>
<PARAG>Em 14 de Junho deste anno foi mandado sahir
immediatamente da Corte o cardeal <NOME>Accioalolli</NOME> e dentro em 4
dias do Reino, para o que se lhe deu decente transporte e acompanhamento.
Era Nuncio de sua Santidade o Papa Clemente 13.</PARAG>
<PARAG>Aqui principiou a rotura com a <LUGAR>Sé Apóstolica</LUGAR>. Fes-se
publico o motivo pela informação que se mandou a <NOME>Francisco
de Almada</NOME> residente na Corte de Roma, dando-se por
fundamento maior o não pôr luminarias por ocasião
do cazamento que se fes no dia seis de Junho do Senhor <NOME>Infante
D. Pedro</NOME> com sua sobrinha a Excelentissima Senhora D. Maria,
princesa do Brasil, o que o dito Cardeal não fizera, porque
lho não fizeram a saber por carta de officio, quando os mais
embaixadores sem isso o fizeram. O caso he que querião
afasta-lo da Corte e tambem elle não seguia a
conduta della
por aquelles tempos.</PARAG></EXCERTO>
```

## Estrutura do documento



## Processamento Estruturado

```
ELEMENT excerto
OUTPUT "%c"

ELEMENT titulo
OUTPUT "<H2>%c</H2>"

ELEMENT parag
OUTPUT "%c<P>"

ELEMENT nome
OUTPUT "<I>%c</I>"

ELEMENT lugar
OUTPUT "<B>%c</B>"
```

Geração da [página HTML](#) correspondente

## Fase de Design

- Análise Documental
  - Desenho da estrutura tentando prever todos os ângulos do problema.
  - Levantamento dos requisitos de aparência visual junto dos futuros utilizadores.
- Resultados
  - Especificação do DTD
  - Especificação do Estilo

## Fase de Edição

- Preparar um editor “especializado”
  - compilar o DTD
  - preparar uma formatação automática para o documento
- Anotar o Documento
  - na sua inserção
  - posteriormente à sua inserção
- Validar o Documento anotado
- Exportar o Documento válido em SGML

## Fase de Validação



- Está integrada nos editores SGML
- É invisível para o utilizador, excepto na sua invocação
- Dá feedback na forma de mensagens que indicam os erros se os houver



## Fase de Formatação ou Transformação



- Processador:
  - Doc. SGML válido + Estilo = OUTPUT
- Processamento
  - Standard
    - DSSSL: jade
    - XSL: saxon, xt, Xalan, ...
  - Pessoal: desenvolvimento de scripts Perl, Omnimark



## Fase de Armazenamento



- O Documento é armazenado juntamente com a sua estrutura
- Tipos de solução:
  - Armazenamento individual usando o sistema de directorias.
  - Bases de Dados orientadas a objectos e desenvolvidas a pensar no SGML: Astoria, InfoManager, Documentor.
  - Implementação sobre Bases de Dados Relacionais baseada em MDA.

