XML (I)

Linguagens de Anotação de Documentos

Informação Estruturada

Linguagens de Anotação

- O LaTeX e o HTML são linguagens de anotação para a escrita e apresentação de *documentos digitais*
  - As anotações identificam diferentes partes estruturais de um documento
- As linguagens de anotação são no entanto usadas para estruturar informação arbitrária, específica para cada *contexto de utilização*
- A “meta-linguagem” mais usada para esse propósito é o XML

Dados Não-estruturados

- No mundo digital queremos partilhar dados entre humanos e entre ferramentas de forma precisa
- Muitos dados são *não-estruturados*: não estão organizados segundo um modelo de dados bem definido
- Isto limita a partilha de informação:
  - Entre humanos, devido a informação implícita e ambiguidades
  - Entre máquinas, porque analisar este tipo de informação requer técnicas inexatas
- E.g., textos em linguagem natural

Dados Não-estruturados

Lista de Grupos

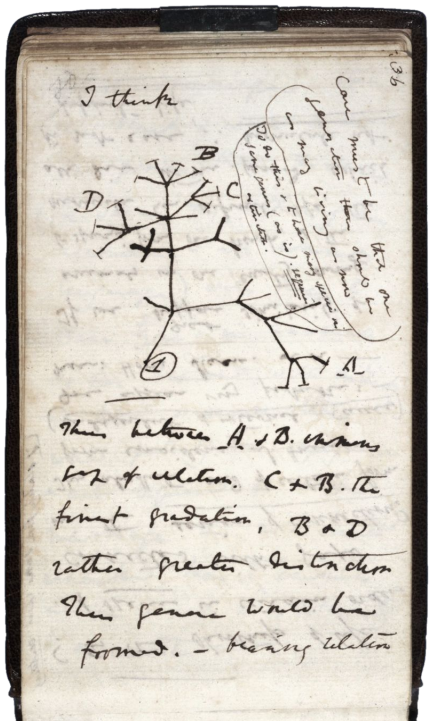
1. Diana Bezerra
   Jessica Magalhães
   Marina Ramos
https://www.overleaf.com/8322246fxpwtwgpvqjv

2. Carolina Carvalho } Tema: Literatura Fantástica
   Cláudia Peças
   Vanessa Silva
https://www.overleaf.com/read/twjyfcdvsqtn
…

6. Catarina Vaz
   Sara Fernandes
https://www.overleaf.com/8322275sgbvpkngrwqk#/29477653/

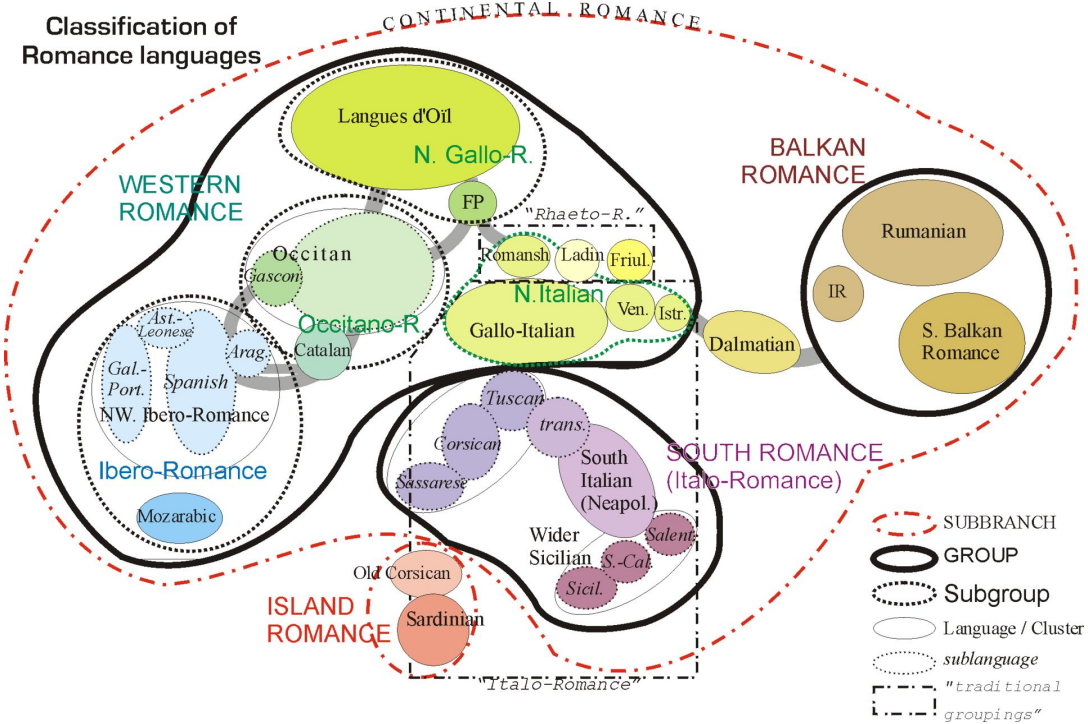
Dados Estruturados

- Os humanos também têm tendência a *estruturar* a informação para a tornar mais legível
  - Diagramas, esquemas, tabelas…
- Modelo conceptual da informação
- Implicam uma *interpretação* dos dados: não é necessariamente única
- Em termos tecnológicos estes modelos têm que ser formalizados para que possam ser processados



Charles Darwin, 1837

Dados Estruturados



Wikipedia

Estruturas de Dados

- *Estruturas de dados* são fundamentais nas ciências da computação
- Além da vantagem conceptual, permitem que dados que sejam *processados* por outras ferramentas
- Algumas funcionalidades básica “grátis”, como fazer *consultas* ou extrair *estatísticas*
    - E.g., “procura todos os alunos com nota positiva”

Estruturas de Dados

- *Tipos* de dados básicos
- Primitivos
  - Números, caracteres, Booleanos (verdadeiro / falso)
- Complexos
  - Combinação de tipos primitivos
  - E.g., um aluno tem nome e número

Estruturas de Dados

- Tipos mais ricos codificam mais informação, mas têm maiores custos de processamento
  - Dados lineares, listas
  - Dados hierárquicos, árvores
  - Dados não-hierárquicos, grafos
- A escolha depende sempre do contexto de utilização

Informação Linear

- *Listas*, sequências (ordenadas) de elementos
  - E.g., texto são listas de caracteres, uma turma é uma lista de alunos, ...
- Listas de listas dão origem a *matrizes*, o conceito subjacente às tabelas e às folhas de cálculo
- Permite estruturar e processar informação facilmente, caso os elementos tenham uma estrutura regular e “lisa”
- Não é adequada quando os elementos têm estrutura variável, ou quando existem relações de um-para-muitos

Informação Linear

id	Aluno1	Aluno2	Aluno3	Tema	LaTeX
1	Diana	Jessica	Marina	NA	http://...
2	Carolina	Cláudia	Vanessa	Literatura Fantástica	http://...
...	...	...	...	...	...
6	Catarina	Sara	NA	NA	http://...

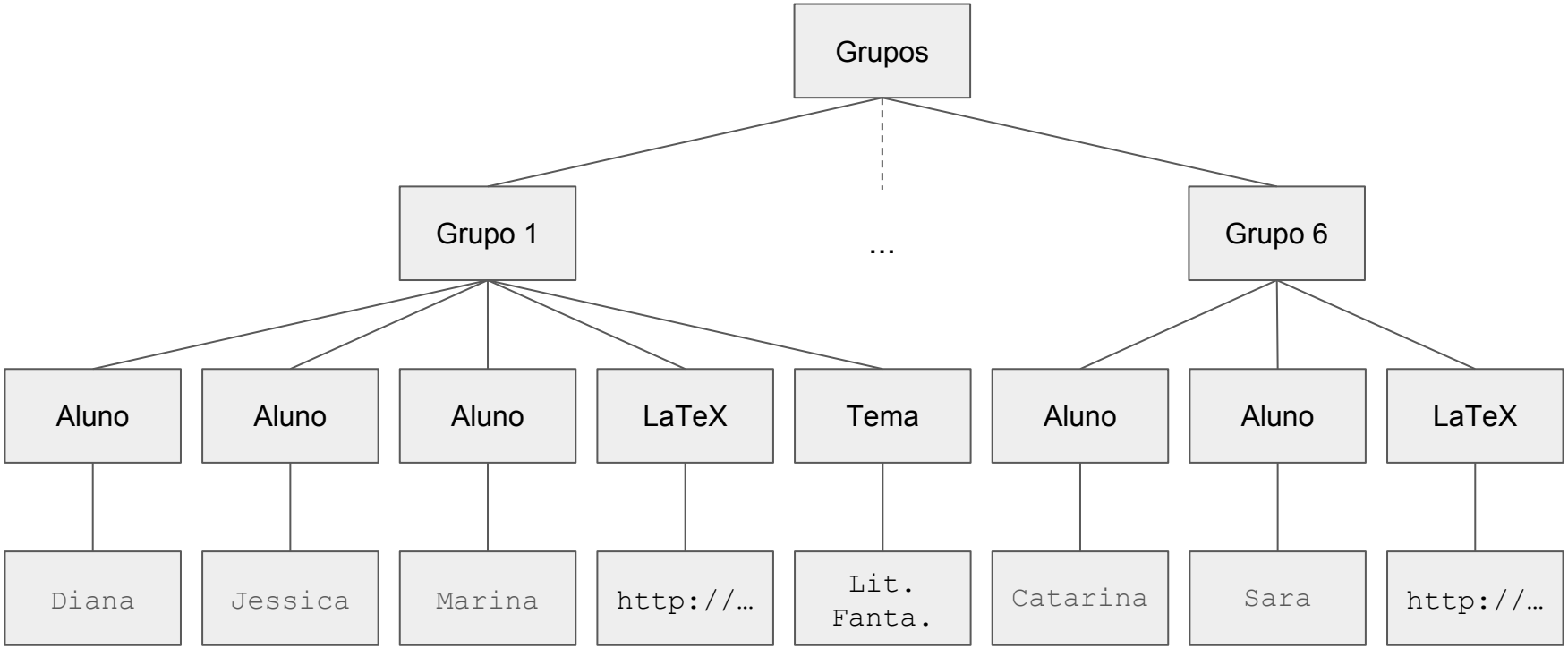
Informação Linear

id	Aluno1	Num1	Aluno2	Num2	...	Tema	LaTeX
1	Diana		Jessica		...	NA	http://...
2	Carolina		Cláudia		...	Literatura Fantástica	http://...
...	...		...		...	...	...
6	Catarina		Sara		...	NA	http://...

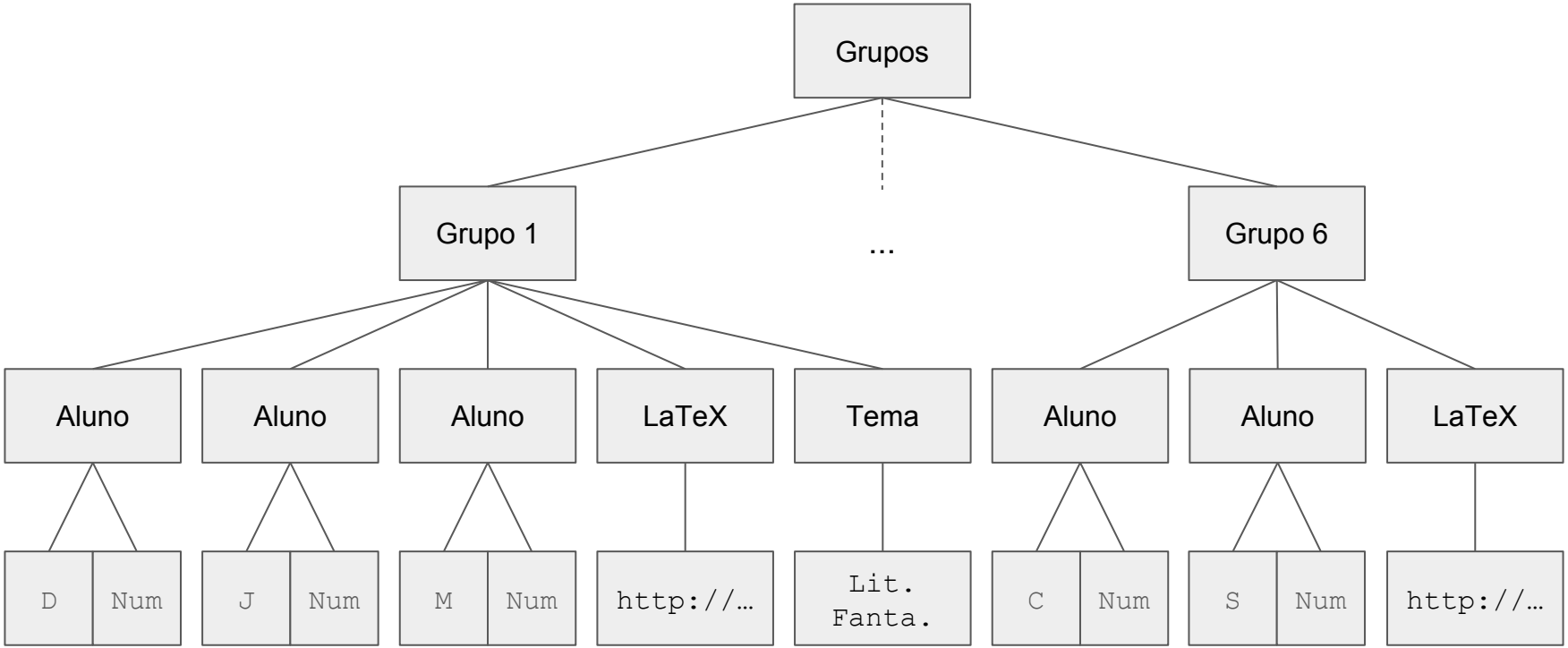
Informação Hierárquica

- *Árvores*, informação hierárquica, agrupada por níveis
- Cada elemento pode ter múltiplos “filhos” (de tipo diferente), e no máximo um “pai”
- Estrutura subjacente às linguagens de anotação mais comuns
  - E.g., um documento contém secções, que contêm sub-secções, ...
- Mais difícil de processar do que informação linear
- Pouco adequada para relações de muitos-para-muitos ou “circular”

Informação Hierárquica



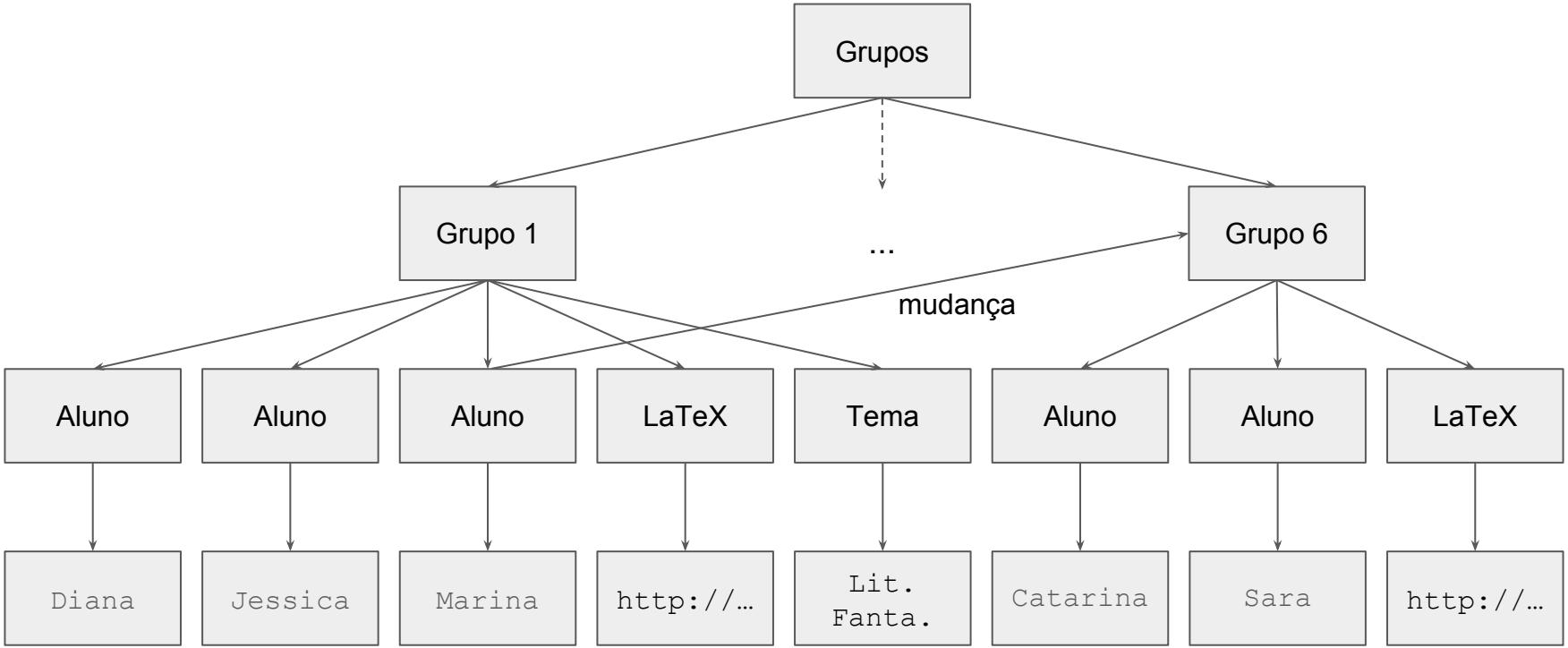
Informação Hierárquica



Informação Não-hierárquica

- *Grafos*, informação não-hierárquica, relações arbitrárias
  - E.g., um aluno está em várias disciplinas e as disciplinas têm vários alunos
- Suporta relações de muitos-para-muitos e circular
- A maior parte dos dados pode ser abstraída para uma estrutura deste género
- Mais complexa de processar (não direcionais, ciclos, ...)

Informação Não-hierárquica



Serialização

- As estruturas de dados são usadas internamente pelas ferramentas, num formato *abstrato*
- Para serem armazenados ou transferidos entre aplicações, os dados têm que ser *serializados* num formato persistente que os permita reconstruir
- Permite abstrair detalhes de *software* e *hardware*
- Particularmente importante na *internet* com processos independentes a comunicar,
- Para que a transmissão de dados seja precisa e sem ambiguidades estes formatos têm que estar formalizados

Serialização

- Muitas vezes queremos que estes formatos sejam legíveis por humanos (apesar de isto ter custos de armazenamento)
- Ou seja, usar linguagens de anotação em texto puro
- Várias tecnologias para este fim
  - XML, JSON, YAML, ...

XML

XML

- O XML (*eXtensible Markup Language*) é uma linguagem de anotações criada para estruturar informação arbitrária
- Ao contrário do HTML, no XML não existe um conjunto de marcas pré-definidas
- Pode ser aplicado em contextos genéricos, desde que as partes interessadas concordem no “esquema”
    - E.g., no contexto das linguísticas, anotações sintáticas, semânticas, fonéticas, ...
- É um acto de interpretação da informação para um contexto de aplicação, implica uma abstração dos conceitos relevantes

XML

- O XML foi proposto (1996) para ser mais flexível que o HTML (elementos de tipo fixo) mas mais fácil de processar que outras linguagens existentes
- Linguagem livre, pode ser utilizada livremente e sem custos
- Regras precisas e padronizadas garantem que os documentos XML podem ser processados de forma consistente e sem erros estruturais
- Tornam o processo também mais eficiente
- Tal como o HTML e o CSS, o *standard* XML é gerido pela W3C

XML

- O XML por si não tem qualquer significado, apenas define as regras pelas quais se podem definir anotações precisas e sem ambiguidade
- Pode ser vista como uma "*meta-linguagem*" que deve ser concretizada pelos utilizadores
- Cada ferramenta processa e manipula os dados como entender
- Completa separação do conteúdo da apresentação: não existem regras para a visualização de XML
- *Extensível*: novos elementos não invalidam os documentos existentes

XML

- Imensos “esquemas” XML estão padronizados para contextos específicos, permitindo partilhar informação consistentemente dentro desses contextos
- Mas os utilizadores são sempre livres de criar novos esquemas, desde que partilhados pelas diferentes entidades
- Formalizar uma taxonomia para um domínio, semântica
    - Uma marca “`grupo`” significa o mesmo para todas as partes; uma marca “`equipa`” não significa nada
- Tecnologias para definir os “esquemas” e para traduções entre “esquemas”

XML

- “*Auto-descritivo*”: os tipos de elementos são definidos ao longo do XML
- A declaração dos tipos de elementos válidos não é obrigatória (mas recomendada quisermos partilhar os dados)
- Olhando unicamente para o documento permite inferir a informação contida

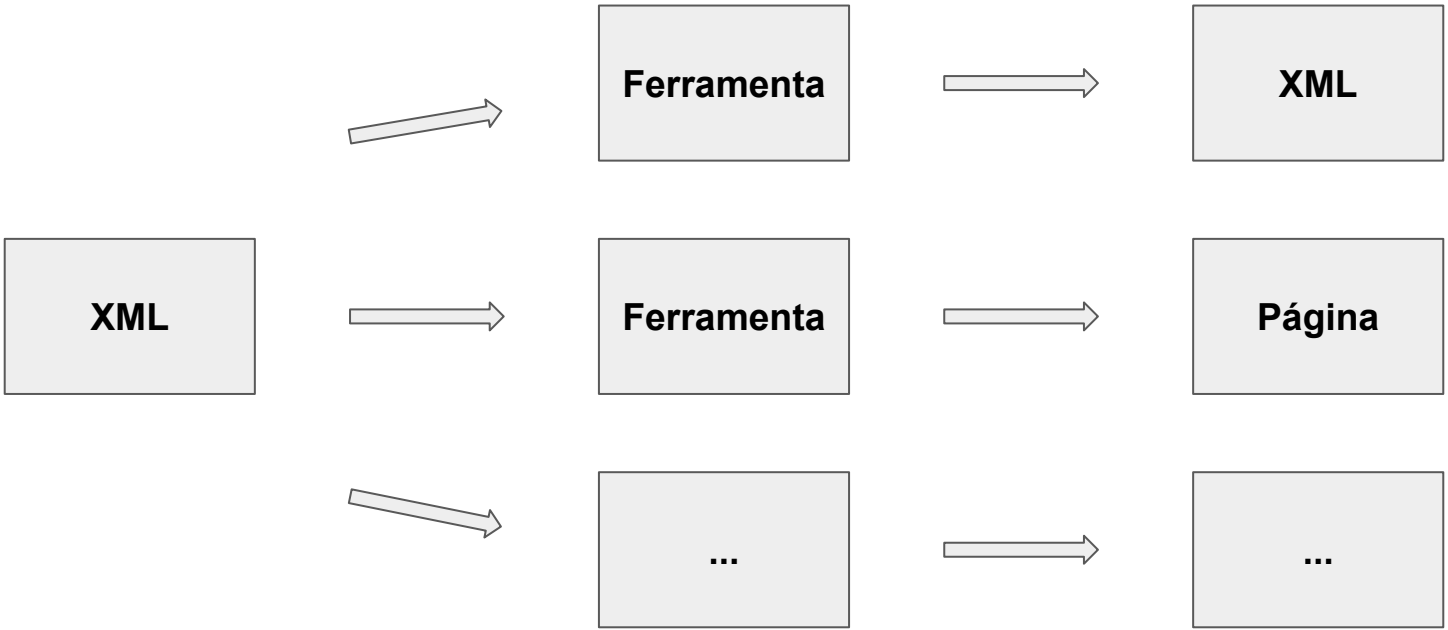
Síntaxe XML

```
<grupos>
     <grupo id="1">
          <aluno>Diana</aluno>
          <aluno>Jessica</aluno>
          <aluno>Marina</aluno>
          <tema>Literatura Fantástica</tema>
          <latex>http://...</latex>
     </grupo>
     ...
     <grupo id="6">
          <aluno>Catarina</aluno>
          <aluno>Sara</aluno>
          <latex>http://...</latex>
     </grupo>
</grupos>
```

XML

XML

XML



Exemplos XML

- Mesmo sem nos apercerbermos, estamos a usar XML constantemente dentro e fora da *web*
- É relevante para quem estiver interessado em criar novo conteúdo, em particular entre serviços na *web*

Exemplos XML

- Notificações na *web*, *Really Simple Syndication*, *rss*

```
<rss version="2.0">
<channel>
  <title>W3Schools Home Page</title>
  <link>https://www.w3schools.com</link>
  <description>Free web building tutorials</description>
  <item>
    <title>XML Tutorial</title>
    <link>https://www.w3schools.com/xml</link>
    <description>New XML tutorial on W3Schools</description>
  </item>
</channel>
</rss>
```

Exemplos XML

- Anotações geográficas, *Keyhole Markup Language*, *kml*

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<Placemark>
  <name>New York City</name>
  <description>New York City</description>
  <Point>
    <coordinates>-74.006393,40.714172,0</coordinates>
  </Point>
</Placemark>
</Document>
</kml>
```

Exemplos XML

- Documentos, *Office Open XML*, *docx*

```
<CoreProperties
xmlns="http://schemas.microsoft.com/package/2005/06/md/core-properties">
   <Title>Word Document Sample</Title>
   <Subject>Microsoft Office Word 2007</Subject>
   <Creator>2007 Microsoft Office System User</Creator>
   <Keywords/>
   <Description>2007 Microsoft Office system .docx file</Description>
   <LastModifiedBy>2007 Microsoft Office System User</LastModifiedBy>
   <Revision>2</Revision>
   <DateCreated>2005-05-05T20:01:00Z</DateCreated>
   <DateModified>2005-05-05T20:02:00Z</DateModified>
</CoreProperties>
```

Exemplos XML

- Documentos anotados, *Text Encoding Initiative*, *tei*

```
<div type="sonnet">
 <lg type="quatrain">
  <l>Les amoureux fervents et les savants austères</l>
  <l> Aiment également, dans leur mûre saison,</l>
  <l> Les chats puissants et doux, orgueil de la maison,</l>
  <l> Qui comme eux sont frileux et comme eux sédentaires.</l>
 </lg>
 <lg type="tercet">
  <l>Leurs reins féconds sont pleins d'étincelles magiques,</l>
  <l> Et des parcelles d'or, ainsi qu'un sable fin,</l>
  <l>Étoilent vaguement leurs prunelles mystiques.</l>
 </lg>
</div>
```

Síntaxe XML

- XML estrutura a informação em árvore
- Existe um elemento que é a *raiz*
- Cada elemento pode conter outros elementos, aninhados
- Estrutura semelhante à do HTML, mas com regras mais restritas

Síntaxe XML

- Um elemento é delimitado por marcas de abertura e de fecho

```
<elemento> … </elemento>
```

- Cada elemento pode conter
  - Texto / conteúdo
  - Elementos “filhos”
  - Atributos
- Elementos sem conteúdo podem ser abreviados

```
<elemento/>
```

Síntaxe XML

- Apenas um elemento de raíz
- Todos os elementos abertos têm que ser fechados
- Elementos têm que estar bem aninhados (o que abre dentro de um elemento tem que fechar dentro desse elemento)
- As marcas são sensíveis à capitalização (`Elemento` diferente de `elemento`)
- Caracteres especiais para símbolos protegidos `&lt` (<) `&gt` (>)
- Estas regras mais restritas facilitam o processamento de ficheiros XML

Síntaxe XML

- Elementos podem conter *atributos*, também definidos pelos utilizadores

```
<elemento atributo="...">
```

- Valores de atributos têm que estar entre aspas
- No HTML era claro o que eram atributos: meta-dados que não seriam apresentados
- No XML nada é apresentado, por isso é mais ambíguo o que devem ser atributos

Síntaxe XML

```
<grupos ano="16/17">

     <grupo id="1">
          <aluno>Diana</aluno>
          <aluno>Jessica</aluno>
          <aluno>Marina</aluno>
          <tema>Literatura Fantástica</tema>
          <latex>http://...</latex>
     </grupo>
     ...
     <grupo id="6">
          <aluno>Catarina</aluno>
          <aluno>Sara</aluno>
          <latex>http://...</latex>
     </grupo>
</grupos>
```

Síntaxe XML

```
<grupos>
     <ano>16/17<ano>
     <grupo id="1">
          <aluno>Diana</aluno>
          <aluno>Jessica</aluno>
          <aluno>Marina</aluno>
          <tema>Literatura Fantástica</tema>
          <latex>http://...</latex>
     </grupo>
     ...
     <grupo id="6">
          <aluno>Catarina</aluno>
          <aluno>Sara</aluno>
          <latex>http://...</latex>
     </grupo>
</grupos>
```

Síntaxe XML

```
<grupos>
     <ano>16/17<ano>
     <grupo id="1">
          <aluno>
               <nome>Diana</nome>
               <numero>...</numero>
          </aluno>
          <aluno>
               <nome>Jessica</nome>
               <numero>...</numero>
          </aluno>
          <tema>Literatura Fantástica</tema>
          <latex>http://...</latex>
     </grupo>
     ...
</grupos>
```

Namespaces

- Como o nome das anotações pode ser livremente definido pelos utilizadores, podem surgir conflitos entre esquemas
  - E.g., misturar XML de “grupos” com XML de “turmas”, diferentes dados de “aluno”
- Nestes casos podemos usar “espaços de nomes” (*namespaces*) para nos referirmos a cada um deles

Namespaces

- *Namespaces* são declarados dentro de qualquer elemento com o atributo

  `xmlns:espaco="http://..."`

- Elementos podem agora ser declarados com esse *prefixo*

  `<espaco:elemento>`

- E.g., grupos vs turmas

```
<unido xmlns:turma="http://..." xmlns:grupo="http://...">
    <turma:aluno>...</turma:aluno>
    <grupo:aluno>...</grupo:aluno>
</unido>
```

Para lá do XML

- Um ficheiro XML é bem formado se seguir as regras definidas atrás
- Mas para que a informação seja consistentemente processada, tem que haver acordo entre o significado das várias anotações
- Os esquemas que definem marcas válidas têm que ser formalizados
- Isto adiciona uma nova camada de validação aos ficheiros XML
- Estes esquemas são definidos com outras tecnologias associadas ao XML, como o DTD e o XSD

Para lá do XML

- Uma das vantagens de usar XML são as tecnologias associadas que permitem consultar e manipular esse tipo de ficheiros
- *XPATH*: navegar e seleccionar elementos num ficheiro XML
- *XQuery*: fazer consultas sobre ficheiros XML
- *XSLT*: definir transformações entre diferentes esquemas XML

SGML

- As similaridades entre HTML e XML são evidentes
- Isto deve-se ao facto de ambas serem “derivadas” da linguagem de anotação SGML (*Standard Generalized Markup Language*)
  - HTML é uma aplicação de SGML (marcas definidas)
  - XML é uma versão restrita de SGML (sub-conjunto de funcionalidades)

SGML

- GML proposta em 1969 pela IBM, uma das primeiras linguagens de anotação
- SGML é uma evolução do GML de 1986, estabeleceu os conceitos gerais por trás do XML
- Tal como o XML, é mais uma meta-linguagem: define as regras para se definirem linguagens de anotação concretas

SGML

- O HTML é a aplicação de SGML mais popular: define um conjunto concreto de elementos
- No entanto algumas limitações técnicas do SGML limitaram a sua adoção
- O XML foi definido como uma versão restrita do SGML, tornando-se mais fácil de utilizar e processar, o que justifica a sua popularidade

Take-home Lesson

- Estruturar informação é essencial para que seja partilhada de modo preciso e sem ambiguidades
- As linguagens de anotação permitirem estruturar dados de acordo com regras restritas, sendo legíveis por humanos e máquinas
- O XML é uma “meta-linguagem”, que pode ser concretizada para qualquer domínio de aplicação
- A sua flexibilidade, associada a regras simples mas precisas, fez com que fosse “universalmente” aceite como o formato de transmissão na *web*

Tutorial

https://www.w3schools.com/xml/