



by Floris Lambrechts  
<floris/at/linuxfocus.org>

## Aprendendo XML



### *About the author:*

Tenho sido, por muitos anos o principal editor da LinuxFocus/Holanda. Estou a estudar "engenharia industrial em electrónica" em Leuven, Bélgica e passo o meu tempo a brincar com o Linux, o PHP, o XML e a LinuxFocus, ao mesmo tempo que leio livros de Stephen Hawking e (pelo momento:) de Jef Raskin, 'The Humane Interface'.

### *Abstract:*

Isto é um pequena introdução ao XML. Conhecerá o Eddy o meta cat, o polícia da sintaxe do XML e algumas DTDs. Não se preocupe nós explicaremos :-)

---

## Introdução

No verão de 2001, alguns dos editores da LinuxFocus vieram juntos, a Bordéus para o LSM. Muito diálogo e discussões acerca da documentação da LSM, o interesse especial do grupo virava-se para a mesma matéria: o XML. Longa (e divertidas) horas decorreram a explicar o que o XML é realmente, o que é bom para sabermos como utilizá-lo. Caso esteja interessado, é isto que este artigo procurará discutir.

Gostaria de agradecer ao Egon Willighagen e ao Jaime Villate por me introduzirem o XML. Este artigo, é de algum modo, baseado na informação dos artigos de Jaime, que pode encontrar nos links abaixo.

## O que é o XML?

Nós, rapazes de documentação, sabíamos o que o XML era, mais ou menos. Apesar de tudo, a sua sintaxe é muito semelhante ao HTML e é outra linguagem de "marcação" como o SGML e (novamente) o HTML, correcto? Sim. Mas existe ainda mais.

O XML tem algumas propriedades que o tornam num formato de dados útil para, praticamente, tudo. Parece que, na maioria das vezes, pode descrever as coisas mais complexas, ao mesmo tempo que permanece de fácil

interpretação para os humanos e fácil de ser interpretador por programas. Como é que é possível? Investiguemos esta linguagem ímpar.

## Eddy, o gato meta

Antes de tudo, o XML é uma *linguagem de marcação*. Os documentos escritos numa linguagem de marcação contêm basicamente, duas coisas: *dados* e *metadados*. Se sabe o que significam os dados, por favor avise-me, mas até então falemos dos metadados :). Simplesmente, dizendo: os metadados são informação extra que adicionam um contexto, ou um significado aos dados em si. Um exemplo simples: repare na frase 'My cat is called Eddy'. Uma pessoa humana como você sabe que 'cat' é o nome de uma espécie de animais e 'Eddy' é o seu nome. Contudo, os programas de computador, não são humanos e não sabem isto. Assim, utilizamos os metadados para dar significado aos dados (claro que é com a sintaxe XML!):

```
<sentence>
  My <animal>cat</animal> is called <name>Eddy</name>.
</sentence>
```

Agora, até mesmo um programa de computador estúpido pode dizer que 'cat' é uma espécie e que 'Eddie' é um nome. Se quiser produzir um documento onde todos os nomes são impressos a azul, e todas as espécies em vermelho, então o XML facilita-nos a tarefa. Só pelo divertimento, eis aqui o que obteríamos:

My cat is called Eddy.

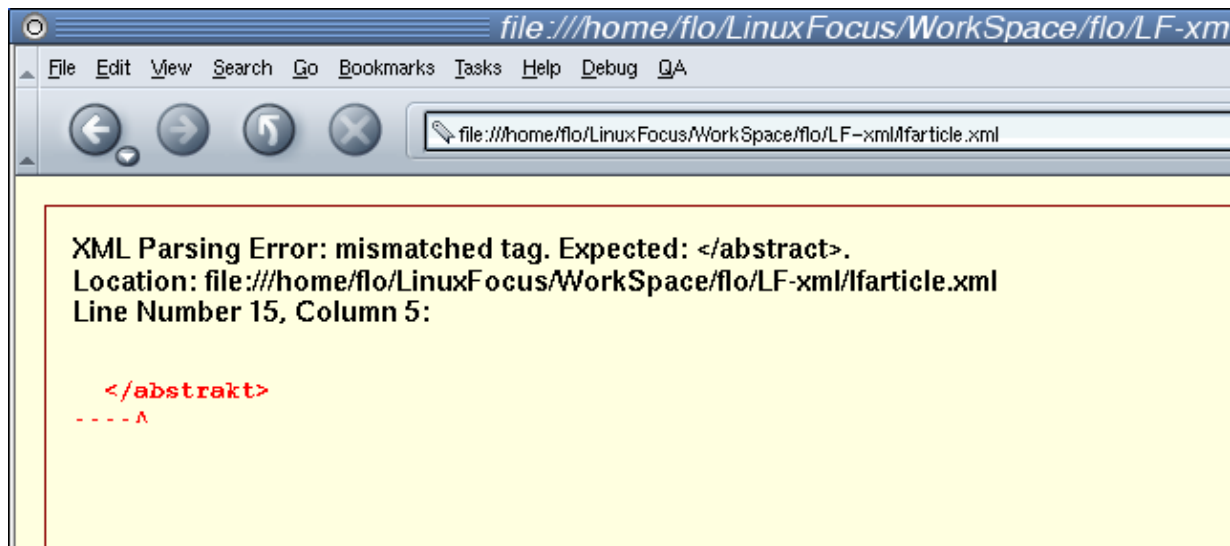
Agora, teoricamente, podemos pôr a informação do layout (as cores neste caso) num ficheiro à parte, um chamado stylesheet. Quando tal fazemos, estamos a separar a informação do layout do conteúdo, algo que é considerado, por alguns, o Cálice Sagrado da Web design™. Até agora, não fizemos nada de especial, adicionar metadados é a função para a qual as linguagens de marcação são desenhadas. Assim, o que é que torna o XML tão especial?

## O polícia da sintaxe

Antes de mais, o XML tem uma sintaxe muito restrita. Por exemplo, no XML toda a <tag> tem de ter uma </tag> de fecho. [ Note: visto que é um pouco estúpido escrever <tag></tag> quando não existe nada entre elas, pode também escrever <tag /> e, eventualmente, aproveitar alguns pares de minutos da sua vida]. Uma outra regra, é que não pode 'misturar' as tags. Temos de fechar as na ordem reversa àquela em que as abriu. Algo como isto não é válido:

```
<B> Texto a Negrito <I>Texto a negrito e itálico </B> texto em itálico </I>
```

As regras de sintaxe dizem que deve fechar a tag </I> antes de fechar a tag </B> E, tenha cuidado, *\_todos\_* os elementos no XML devem estar contidos em tags (claro com excepção das tags externas!). Isto, é a razão pela qual no exemplo acima, nós escrevemos as tags <sentence> à volta da frase. Sem elas, algumas das palavras na frase não seriam incluídas entre as tags, e isto, como outras imensas coisas, tornam o polícia da sintaxe do XML realmente muito mau.



### A Política de sintaxe do Mozilla a trabalhar...

Mas uma força de polícia forte tem as suas vantagens: traz ordem. Como o XML segue uma sintaxe restrita, é muito fácil para os programas lê-la. Os dados nos seus documentos XML, também se encontram muito estruturados, o que os torna de fácil leitura e escrita para humanos.

Por favor, note que qualidade 'teórica' do XML nem sempre pode ser posta em prática. Por exemplo, a maioria dos parsers de XML correntes estão longe de ser rápidos, e por vezes são, realmente, enormes. Assim parece que o XML, não é de tão fácil leitura para o software em geral. Digamos que, não é uma boa ideia fazer \*tudo\* em XML, só porque pode. Para aplicações onde precisa de fazer várias pesquisas num documento, ou onde tem documentos, realmente, enormes, o XML não é frequentemente a escolha certa. Mas tal não significa que é impossível utilizar o XML com estes propósitos.

Um exemplo simpático e que demonstra o poder do XML, mas também a sua lentidão, é o facto de poder escrever base de dados nele (tente fazê-lo com o HTML! :p). Foi exactamente o que o Egom Willighagen fez para a secção alemã da LinuxFocus, o seu artigo acerca deste sistema está disponível nos links no fim desta página. Neste caso a flexibilidade e extensibilidade de um formato de ficheiro artesanal foi escolhido ao invés da velocidade (diga-se MySQL).

Respeitante à sintaxe restrita do XML: se pretende ficar bom amigo dos verificadores de sintaxe, então existem várias maneiras de fazer com que o polícia da sintaxe faça algum do seu trabalho. Caso o pretenda, tem de fazer uma utilização inteligente da DTD...

## A DTD

No nosso pequeno exemplo, acima, 'Eddy o gato-meta' inventámos as nossas próprias tags de XML. Claro que tal acto criativo não é tolerado pelo polícia da sintaxe! O 'homem de azul' quer saber o que está a fazer, como, quando e (se possível) porquê. Bem, sem problemas, pode explicar tudo com a DTD...

Uma DTD permite-lhe 'inventar' novas tags. De facto, permite-lhe inventar novas linguagens, desde que sigam a sintaxe do XML.

A DTD, ou **D**ocument **T**ype **D**efinition (Definição do tipo de Documento), é um ficheiro que contém uma descrição da linguagem do XML. É, actualmente, uma lista de todas as tags possíveis, dos seus possíveis atributos, e das suas possíveis combinações. A DTD descreve o que é possível na sua linguagem XML e o que não é. Assim quando falamos da 'linguagem XML', estamos, na realidade, a falar de uma DTD específica.

## Ponha o polícia a trabalhar

Por vezes a DTD *obrigá-lo-á* a fazer algo num determinado sítio. Por exemplo, a DTD pode força-lo a incluir uma tag que contenha o título do documento. O que existe de bom nisto, é que existe software actual (por exemplo um módulo do emacs) que escreve as tags requeridas automaticamente.

Deste modo, algumas partes da estrutura dos seus documentos são preenchidas automaticamente. Por causa da sintaxe ser tão restrita e bem definida, a DTD pode guiá-lo através do processo de escrever um documento. E quando comete erros, como esquecer-se de uma tag de fecho, o polícia informa-a. Assim, ao fim de contas, os polícias não são tão 'maus'; onde os polícias do mundo real dizem 'Tem o direito de permanecer em silêncio' o polícia do XML, diz-lhe muito amigavelmente, acerca de um 'Erro de sintaxe na linha xx : '... :) E enquanto o polícia faz todo o seu trabalho, claro que *\*você\** se pode concentrar somente no conteúdo.

## Na Remistura

Uma última grande característica do XML, é a habilidade de utilizar várias DTDs ao mesmo tempo. Isto quer dizer que pode utilizar vários tipos de dados diferentes, ao mesmo tempo, no mesmo documento.

Este 'mistura' é feita com os namespaces do xml. Por exemplo, você pode incluir a DTD Docbook no seu documento .xml (neste exemplo com o *prefixo* 'dbk').

Todas as tags do Docbook estão prontas a ser utilizados no seu documento na seguinte forma (suponha que existe a seguinte tag no Docbook, <just\_a\_tag>):

```
<dbk:just_a_tag> somente algumas palavras </dbk:just_a_tag>
```

Utilizando o namespace system, você pode utilizar qualquer tag de qualquer DTD do xml. Abre um mundo de possibilidades, como pode ver no próximo capítulo...

## DTDs Disponíveis

Aqui está uma pequena colecção de DTDs que estão (em parte) em utilização.

- **Docbook-XML**

O Docbook é a linguagem para escrever documentos estruturados, por exemplo, livros e papeis. Mas, também é utilizada para outras tarefas diferentes. O Docbook é, actualmente, uma DTD SGML (o SGML é um standard de marcação), mas existe, também, uma versão popular em XML. Esta é uma das mais populares DTDs do XML.

- **MathML**

O MathML é a linguagem de marcação Matemática, que é utilizada para descrever expressões e fórmulas matemáticas. É, realmente, um utilitário elegante para as pessoas do mundo da matemática. Os Químicos, por seu lado, não precisam de ter inveja dos seus colegas matemáticos, para eles existe algo como a CML, ou Chemical Markup Language (Linguagem de Marcação Química). Note que o Mozilla 1.0 traz, agora suporte para a MathML, por omissão.

- **RDF**

A RDF é a Resource Description Framework. É desenhada para ajudar a codificar e a reutilizar os metadados; na prática é utilizada por muitos sites para dizer a outros as notícias que estão a apresentar. Por exemplo, o site Alemão [linuxdot.nl.linux.org](http://linuxdot.nl.linux.org) utiliza o ficheiro RDF de outros sites para apresentar os itens de notícias. A maioria dos sites populares de notícias (como por exemplo a Slashdot) têm ficheiros RDF disponíveis para que possa copiar os títulos das suas notícias para uma sidebar da sua página pessoal, por exemplo.

- **SOAP**

O SOAP significa Simple Object Access Protocol (Protocolo simples de acesso a objectos). É uma linguagem utilizada por processos para comunicar uns com os outros (trocar dados e executar chamadas a procedimentos remotos). Com o SOAP os processos podem comunicar remotamente uns com os outros, por exemplo sob um protocolo http (internet). Penso que aqui o Atif da LF lhes pode dizer mais acerca disto, veja os links :-)

- **SVG**

Scalable Vector Graphics. O trio PNG, JPEG2000 e SVG é suposto englobar o futuro das imagens na web. O PNG tomará as regras do GIF (bitmaps o mais comprimidos possível com transparência), e o JPEG2000 poderá suceder o .jpg de hoje (bitmaps com um grau configurável de compressão). O SVG não se baseia em bitmaps, mas é um formato de imagem à base de vectores, o que quer dizer que as imagens não são representadas por pixels, mas por formas matemáticas (linhas, quadrados,...). O SVG tem também funções como scripting e animação, assim, neste sentido, pode compará-la com o Flash da Macromedia. Pode utilizar JavaScript nos ficheiros .svg, e utilizando o JavaScript pode, por sua vez escrever código .svg. Muito flexível, não é?

Mas o svg é relativamente novo; de momento, só existe um plugin SVG da Adobe para browsers, de alta qualidade, para as plataformas Windows & Mac. O Mozilla está a trabalhar num visor embebido SVG, Mas este ainda não está completo e tem de obter uma versão especial, compilada do browser para o poder utilizar.

*NOTE:* os ficheiros .svg podem tornar-se enormes e, isto é o porquê de muitas vezes ir encontrar ficheiros .svgz. Existem versões comprimidas utilizando o algoritmo do gzip.

- **XHTML**

XHTML é a variante XML da versão do HTML 4.01. Devido à sintaxe restrita do XML, existem algumas coisas que pode fazer no HTML mas que não são válidas no XHTML. Mas por outro lado, uma página que escreva em XHTML é ao mesmo tempo uma página HTML válida. Note que o programa tidy HTML pode converter as suas páginas existentes em HTML para XML.

- **Os outros**

Muitos novos formatos utilizam o XML, muitas vezes combinados com a compressão .gz ou .zip. Somente um exemplo, o KOffice utiliza os formatos de ficheiro DTD's do XML. Isto é muito útil, por que permite a combinação de funcionalidade de diferentes aplicações num só documento. Por exemplo, pode escrever um documento KWord, com um folha de cálculo KChart embutida nele.

## Links

O W3C, ou World Wide Web Consortium

Eles têm informação acerca do XML, da MathML, da CML, do RDF, do SVG, do SOAP, do XHTML, dos namespaces...

[www.w3.org](http://www.w3.org)

Algum material de Jaime Villate (pode precisar de um tradutor online para ler os dois primeiros:)

[Introdução ao XML](#) (em Espanhol)

[Como gerar HTML com o XML](#) (em Espanhol)

[LSM-slides](#)

O programa, HTML tidy:

[www.w3.org/People/Raggett/tidy](http://www.w3.org/People/Raggett/tidy)

O Docbook  
[www.docbook.org](http://www.docbook.org)

O projecto SVG do Mozilla  
[www.mozilla.org/projects/svg](http://www.mozilla.org/projects/svg)

Artigos da LinuxFocus relevantes:  
[Utilizando o XML e o XSLT para construir a LinuxFocus.org\(/Nederlands\)](http://LinuxFocus.org(/Nederlands))  
[Construindo documentos PDF com o Docbook](#)

---

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Floris Lambrechts "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a></p>	<p>Translation information: en --&gt; -- : Floris Lambrechts &lt;floris/at/linuxfocus.org&gt; en --&gt; pt: Bruno Sousa &lt;bruno/at/linuxfocus.org&gt;</p>
---	---

2005-01-10, generated by lfparsr\_pdf version 2.51