

Tema1

XML

Contidos

Artigos

XML	1
Linguaxes de marcas	1
Codificaci3n de documentos	2
XML:A linguaxe XML	4
XML:O documento XML	7
XML:Manipulaci3n de arquivos	9
XML:Validaci3n de documentos	9
XML:Especificaci3ns relacionadas co XML	10
XML:Ambitos de aplicaci3n	10
Software para edici3n de XML	11

Referencias

Fonte dos artigos e contribuyntes	14
Fontes das imaxes, licenzas e contribuyntes	15

Licenzas de artigos

Licenza	16
---------	----

XML

As linguaxes de marcas serven para codificar documentos a través de marcas textuais (tamén chamadas etiquetas), de maneira que se engade certa información propia información do documento. Esta nova información pode referirse a súa estrutura, presentación, significado, etc.

Víxase a evolución histórica das linguaxes de marcas e a súa clasificación.

Linguaxes de marcas

Historia das linguaxes de marcas. SGML e XML: HTML e XHTML

O concepto da separación entre a estrutura e a presentación xurdiu varias veces nos anos 60 e 70 coa creación de múltiples linguaxes de marcas como *Scribe* ou *TeX*.

Pero sería a IBM quen creou a linguaxe GML, que posteriormente, no ano 86, se estandarizaría no SGML (*Standard Generalized Markup Language*). Falando con propiedade, o SGML non é unha linguaxe senón que é unha metalinguaxe, é dicir, serve para construír linguaxes de marcas concretas para almacenar tipos de documentos reais.

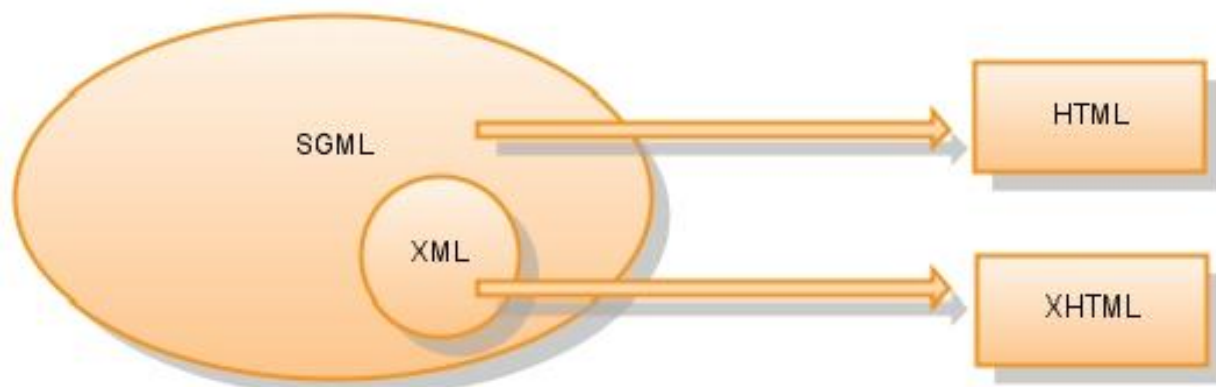
A importancia do SGML é tal, que serviría de base para a creación de multitude de linguaxes específicas de marcas. O máis importante deles, con diferenza, creouse a principios dos anos 90, e foi a primeira gramática da linguaxe HTML (e ata a versión 4.01).

O problema do SGML é que, debido aos ambiciosos obxectivos suscitados á hora da súa creación, se converteu nunha linguaxe moi potente e complexa, e polo tanto moi difícil de aprender e de utilizar. Ademais tiña moitas características pensadas para optimizar a escritura manual dos documentos no ordenador (por exemplo, a posibilidade de non pechar ou omitir etiquetas), o cal complicaba a súa interpretación e aplicacións.

Por iso se creou o XML (*eXtensible Markup Language*), que non é máis que un subconxunto do SGML creado para simplificar a creación das gramáticas de linguaxes de marcado.

Os deseñadores do XML deixaron fóra as partes menos utilizadas do SGML, conseguindo que a especificación do XML ocupe 30 páxinas, fronte ás 500 páxinas do SGML. Segundo Richard Ligth, no seu libro "*Presenting XML*", XML ofrece o 80% das vantaxes do SGML cun 20% da súa complexidade.

Igual que HTML se creou en base a SGML, o XHTML (*eXtensible HyperText Markup Language*) creouse en base a XML. Por iso, o XHTML é a linguaxe pensada para substituír o HTML.



A versión XHTML 1.0 ten as mesmas funcionalidades que o HTML pero cumprindo as estritas especificacións do XML. A versión 1.1 e futuras versións intentarían esquecerse das clásicas etiquetas HTML para finalmente lograr unha web semántica, onde realmente a información e a súa representación estarian separadas.

Codificación de documentos

Sistemas de codificación

A codificación ou *encoding* dun documento fai referencia ao xogo de caracteres empregado no mesmo para representar os símbolos que contén.

Xogos de caracteres

Os primeiros xogos de caracteres que se empregaron en informática foron os seguintes:

- † ASCII (*American Standard Code for Information Interchange*). Emprega os primeiros 7 bits dun byte. Defínese como se debían representar os números, as letras (fala inglesa soamente), signos de puntuación e caracteres de control. O oitavo bit usábase para o control da paridade.
- † Variantes de ASCII (ASCII estendido). Para poder representar novos caracteres, xurdiron diversas variantes de ASCII. Aínda que algunhas tamén modificaban parte do espazo orixinal (concretamente o asignado aos caracteres de control), todas basébanse en eliminar o bit de paridade e aproveitar o espazo entre 128 (10000000) e 255 (11111111). Algunhas das máis coñecidas son:
 - † CP437. É o xogo de caracteres orixinal do IBM PC (o que viña na ROM dos adaptadores de vídeo). Inclúe vocais minúsculas con tilde, eñes e apertura de exclamacións e interrogacións. Non inclúe, por exemplo, vocais maiúsculas con tilde, salvo a Ñ.
 - † CP850. É unha adaptación do CP437 que se empregou en MS-DOS en países de Europa occidental. Inclúe, entre outros, todas as vocais minúsculas con tilde.
 - † Windows Code Pages. Son xogos de caracteres empregados polos sistemas operativos Windows na década dos 80. Existen varios, cada un pensado para ser empregado nun alfabeto determinado. En Europa occidental e países de lingua inglesa empregouse o xogo de caracteres Windows-1252.
 - † ISO-8859. En realidade son varios xogos de caracteres, moi empregados en sistemas Linux, que amplían ASCII utilizando 8 bits e deixando intactos os 128 primeiros caracteres orixinais. O máis usado é o ISO-8859-1, pensado para o alfabeto inglés e outras linguaxes de Europa occidental. ISO-8859-15 é unha pequena variante que inclúe o símbolo do euro (€). Segue a usarse amplamente hoxe en día.
 - † CP858. Pequena modificación ao CP850 para incluír o símbolo do euro.

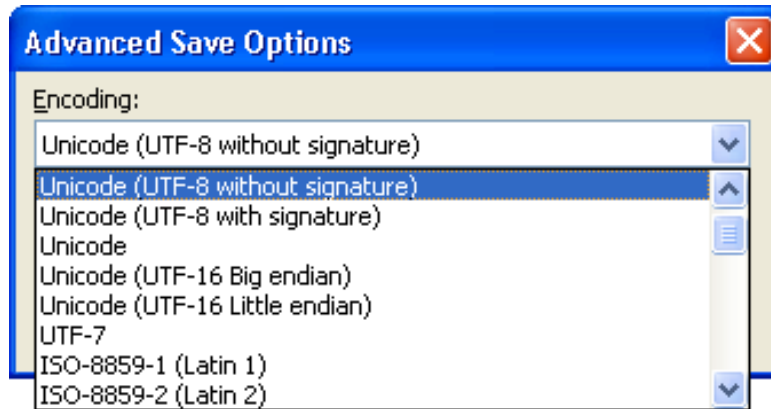
Unicode

Para resolver este caos creouse o estándar Unicode. Consiste nun repertorio de máis de 100.000 caracteres que abranguen a maior parte de alfabetos empregados na actualidade. Existen diversas formas de especializar a codificación de cada carácter. As máis utilizadas son:

- † UTF-8. Emprega codificación variable. Os 128 caracteres do ASCII orixinal codifícanse nun byte que sempre comeza por cero. Para representar outro carácter de ampla utilización (vocais con tilde, eñes, alfabeto grego, etc.), empréganse 2 bytes. Os demais caracteres usan 3 ou 4 bytes. É moi utilizado pois posibilita a codificación de calquera carácter Unicode reducindo considerablemente o espazo ocupado polos documentos se o comparamos coas solucións alternativas.
- † UTF-16. Emprega tamén codificación variable. Neste caso, divide os caracteres en dous grupos: os de máis ampla utilización, que codifica en 2 bytes, e o resto, que ocupan 4 bytes cada un.
- † UTF-32. É unha codificación de lonxitude fixa que utiliza 4 bytes por carácter.

Conv, n ter sempre en conta o seguinte:

- ‡ **f moi importante especificar o xogo de car•cteres que utilizamos cando creamos un documento.** Cando un documento non indica o xogo de car•cteres que emprega, cada aplicaci3n o representar• no que considere m•is adecuado ou no que teña configurado como xogo de car•cteres por defecto. Os navegadores web, por exemplo, perm•tennos consultar y definir o xogo de car•cteres que utilizan para representar cada documento.
- ‡ **Cando creamos un documento „ XML, debemos almacenalo utilizando o xogo de car•cteres que indiquemos na s„a declaraci3n.** Isto „, cando indiquemos que un documento utiliza UTF-32, teremos que almacenalo empregando UTF32. Deberemos observar que o espazo en disco ocupado sexa aproximadamente o cu•druplo que si utilizamos UTF-8 ou ISO-8859-15.



- ‡ A„nda cando esteamos representando un documento no mesmo xogo de car•cteres no que foi creado, isto non garante que a representaci3n en pantalla sexa correcta. Necesitamos ademais que o tipo de letra que esteamos usando conteña t3dolos car•cteres que utiliza o documento. A gran maior„a de tipos de letra conteñen soamente os car•cteres m•is habituais que utilizamos ao escribir documentos.

XML:A linguaxe XML

A linguaxe XML

A linguaxe XML (*eXtensible Markup Language*) creouse para estruturar, almacenar e transportar información. Un documento en XML non é máis que información textual acoutada por etiquetas.

Un documento XML non ten etiquetas preestablecidas. XML permite ao autor do documento definir a súa estrutura e crear as súas propias etiquetas. Por iso diase que o XML (do mesmo xeito que o seu pai, o SGML), é unha metalinguaxe, pois serve para crear linguaxes que almacenan a información que necesita unha aplicación concreta.

As distintas aplicacións que utilicen estes arquivos poden acceder a devandita información e xestionala como lle interese: enviala, transformala, visualizala, etc.

Diase que é extensible porque se poden incorporar novos contidos aos documentos (novas etiquetas ou atributos) sen ter que cambiar as aplicacións que os utilizan.

Exemplo: <toggledisplay> Un exemplo de documento XML podería ser o seguinte:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE pedidos SYSTEM "pedidosEmpresa.dtd">
<!-- comentario no documento -->
<pedidos>
  <pedido num="1">
    <cliente>345</cliente>
    <data>2009/1/15</data>
    <li>
      <producto>325</producto>
      <cantidad>2</cantidad>
      <precioUnitario>1,35</precioUnitario>
    </li>
    <li>
      <producto>27</producto>
      <cantidad>1</cantidad>
      <precioUnitario>32,50</precioUnitario>
    </li>
  </pedido>
  <pedido num="2">
    <cliente>24</cliente>
    <data>2009/01/17</data>
    <li>
      <producto>73</producto>
      <cantidad>1</cantidad>
      <precioUnitario>7,10</precioUnitario>
    </li>
    <li>
      <producto>63</producto>
      <cantidad>9</cantidad>
      <precioUnitario>2,80</precioUnitario>
    </li>
  </pedido>
```

```
</pedi dos>
```

```
</toggledisplay>
```

Os elementos: etiquetas, atributos e contido

As partes que compoñen un documento XML son os elementos, e fñrmanse con etiquetas: textos recotecidos polo analizador e delimitados polos caracteres < >.

Os elementos poden ter contido (outros elementos, texto ou ambos), e poden ser elementos baleiros. Opcionalmente tam, n poden incorporar pares atributo="valor".

```
<nomeEtiqueta [atributo1="valor1"] [atributo2="valor2"] ... >
  contido
```

```
</nomeEtiqueta>
```

elemento con contido

```
<nomeEtiqueta [atributo1="valor1"] ... />
```

elemento baleiro

Exemplo: <toggledisplay>

```
<titulo>O meu primeiro libro de pesca</titulo>
```

```
<cantidad unidades="kg">7</cantidad>
```

```
<asignatura numHoras="12">Programaci3n en
```

```
CSharp</asignatura>
```

```
<pausa/>
```

```
</toggledisplay>
```

A.nda que cada documento XML ten os seus propios elementos, atributos e estrutura axeitada a sfa finalidade, existe un atributo especial que se soe utilizar en todos eles cando se quere indicar o idioma no que est3 descrito o contido. Este atributo denomínase **xml:lang** e o seu valor , o c3digo de dfa letras que identifica o idioma (es, fr, etc.).

```
<descripci3n xml:lang="es"> ... </descripci3n>
```

Nun documento XML, os elementos deben cumprir unhas normas moi estritas:

- † Poden anitarse, pero se pechar3n estruturadamente, , dicir, en orden inversa a sfa apertura.
- † Os elementos sen contido poderanse abrir e pechar no mesmo momento, indicando na etiqueta o smbolo / antes do smbolo de peche (<etiqueta />). Tam, n se pode abrir primeiro e pechar a continuaci3n, o que non se permite en ningfn caso , unicamente abrilos.
- † Un atributo non pode aparecer m3is dunha vez nun elemento, a.nda que se poden especificar en calquera orde.
- † Os valores dos atributos ir3n sempre delimitados por comi3as (simples ou dobres)
- † Todos os elementos do documento an...tanse en forma de 3rbore, existindo un fnico elemento ra3z, do que colgan todos os demais.
- † XML define un conxunto de caracteres como "espazos en branco" (espazos (32), tabuladores (9), saltos de li3a (10) e retornos de carro (13)), que os procesadores permiten para facer m3is lexible o c3digo (desprazar as etiquetas para reflectir o anitamento).
- † Os documentos XML poden conter comentarios. Estes aparecer3n sempre entre as marcas de inicio e peche de comentario, respectivamente.

Con respecto 3s nomes dos elementos e atributos:

- ‡ Comezaran sempre por unha letra.
- ‡ Non poden comezar por XML nin variantes de maiúsculas/minúsculas.
- ‡ No poden conter espazos.
- ‡ Distinguen entre maiúsculas e minúsculas (son case-sensitive).

Cando un documento XML cumpre estas normas dise que está **ben formado**.

As entidades

As entidades son estruturas XML cun nome asociado. Para referenciar unha entidade faise coa seguinte sintaxe:

```
&nomeDaEntidade;
```

Ao referenciar unha entidade polo seu nome inséase automaticamente o seu contido no canto da súa referencia.

Posteriormente comentaremos cómo crear entidades, pero agora veremos qu, entidades están predefinidas:

Carácter	Referencia
&	&
<	<
>	>
"	"
'	'
	&#codChar;

Moitos veces o uso das entidades pode ser moi incómodo, e a lectura do documento moi complicada para os "humanos". Por iso en ocasións pode ser moi útil utilizar unha sección `<![CDATA[...]>` posto que o parser XML ignora o seu contido (algo similar á etiqueta `<pre>` de [[HTML]]).

Exemplo: `<toggle>` O XML seguinte:

```
<code>&lt; H1 &gt; O operador de concatenación:
& &lt; /H1 &gt; </code>
```

Pódese substituír por:

```
<code><![CDATA[<H1>O operador de concatenación:
&</H1>]]></code>
```

```
</toggle>
```

Espazos de nomes (*namespaces*)

Os espazos de nomes son unha recomendación W3C (*World Wide Web Consortium*) ^[1] para que os nomes comúns de elementos e etiquetas non colisionen, ben porque provén de distintos documentos XML a un posible documento destino, ben porque nos interesa separar dun xeito estruturado o contido dun documento.

^ dicir, se a cada un dos elementos ou atributos asignáse un espazo de nomes (onde se deben ser únicos), no caso de combinalos non colisionarán. Así, por exemplo, poderemos ter unha mesma etiqueta chamada `<endereco>` para clientes e provedores e non se confundirán posto que cada unha estará precedida polo seu espazo de nomes.

A definición dun espazo de nomes realízase a través do atributo `xmlns` cuyo valor será un identificador global único (URI): adoita pórse un URL (que, un tipo de URI) pero tratado como cadea, non se comproba a súa existencia).

Pódese referenciar un espazo de nomes para unha etiqueta e o seu contido definíndoo directamente como atributo na propia etiqueta:

Deste xeito, cada etiqueta irá precedida do seu espazo de nomes. Pero isto fai que a lexibilidade dos documentos se vexa seriamente comprometida, posto que as etiquetas poden ter a forma `<http://www.xunta.es/cursos:directión>`. Para elo crearemos un prefixo máis curto utilizando o atributo `xmlns` da forma `xmlns:prefixo="valor URI"`.

```
<xg: cursos xmlns:xg="http://www.xunta.es/cursos">
  <xg: curso nome="C#">
    <xg: alumno ...
    ...
  </xg: alumno ...
  ...
</xg: curso>
...
</xg: cursos>
```

Os distintos espazos de nomes utilizados nun documento adóitanse definir no elemento raíz con tantos atributos `xmlns` como sexan necesarios. Se un atributo `xmlns` non define un prefixo senon que simplemente ten un valor: `xmlns="http://www....."`, pois as etiquetas sen prefixo contidas nese elemento referiranse a ese espazo de nomes (por defecto).

Referencias

[1] <http://www.w3c.es/>

XML:O documento XML

O documento XML

O documento XML consta do prólogo (opcional) e do corpo. O prólogo pode constar da declaración XML, a declaración de tipo de documento e instrucións de procesamento. O corpo forma o documento en si, é dicir, as etiquetas e o seu contido.

O prólogo e a declaración XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

Os atributos son:

- ‡ `version`: A versión do XML utilizada (habitualmente 1.0, xa dispoñible a 1.1).
- ‡ `encoding`: Reférese ao sistema de codificación utilizado.
- ‡ `standalone` (yes | no) : informa de se o documento está aillado, ou sexa, se non depende de declaracións externas (como un DTD, p.e.).

O prólogo é o tipo de documento

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final/EN">
<!DOCTYPE documento SYSTEM "doc.dtd">
<!DOCTYPE clientes SYSTEM
"http://www.servidor.com/defs/clientes.dtd">
<!DOCTYPE elementoRaz [
definição do DTD
]>
```

DOCTYPE especifica o nome do elemento *raz* do documento e, a continuacão, o tipo de documento mediante alguns dos seguintes elementos:

- ‡ PUBLIC , o identificador público que faz referência ao DTD.
- ‡ SYSTEM , um URI (Identificador Universal de Recursos) localizando a rota do DTD.
- ‡ o DTD integrado no próprio documento XML.

O prólogo é instruções de processamento

As instruções de processamento podem-se utilizar para passar informação às aplicações e informá-las de como administrar os dados do documento. Ainda que adótem a forma de prólogo, podem situar-se em qualquer lugar do documento (fora dos outros elementos de marcado).

A sua sintaxe , *<?fonte instruções?>*, onde *fonte* , o nome da aplicação que interpreta as instruções. O seu nome não pode começar por nenhuma combinação de maiúsculas/minúsculas de "xml" posto que estas são reservadas pelos padrões XML.

Exemplos de instruções de processamento muito utilizadas são:

- ‡ *<?xml-stylesheet ... ?>* para aplicar uma folha de estilo ao documento.
- ‡ *<?php ... ?>* para adicionar código PHP

Por exemplo, se queremos usar uma folha de estilo CSS para apresentar o conteúdo do documento adicionaríamos a seguinte linha:

```
<?xml-stylesheet type="text/css" href="estilos.css"?>
```

Falaremos em posteriores capítulos das diferentes possibilidades de como apresentar o conteúdo do documento XML.

O corpo

Consta das etiquetas XML necessárias para estruturar o documento junto ao seu conteúdo. Lembrar que existem, pelo menos, um elemento *raz*, para que o documento seja considerado como bem formado.

XML:Manipulación de arquivos

Manipulación de arquivos XML

A creación dun arquivo XML pódese realizar cun simple editor de textos, aínda que, preferible (dada a riqueza sintaxe que implica) utilizar un editor de XML que facilite a tarefa, ou simplemente unha aplicación que o exporte doutra fonte de datos.

Existe unha gran variedade de aplicacións específicas para levar a cabo edición de arquivos XML.

Tan importante como a creación, a manipulación de ditos arquivos. As aplicacións que utilizan documentos XML a miúdo utilizan APIs xa existentes no mercado.

Dúas APIs abertas e independentes da linguaxe son DOM e SAX.

- ‡ DOM (*Document Object Model*), unha representación en memoria da estrutura do documento XML preparada para acceder e modificar os seus obxectos, que representan os elementos e atributos do documento.
- ‡ SAX (*Simple API for XML*) está baseado na xeración de eventos cada vez que atopa un novo elemento XML ou un erro no documento. É responsabilidade da nosa aplicación capturar ditos eventos e actuar en consecuencia.

XML:Validación de documentos

Validación de documentos

A validación dun documento realízase mediante o uso doutro documento que nos permita especificar a estrutura dun tipo de documento XML concreto para unha aplicación específica. En dito documento definimos a súa estrutura e sintaxe, é dicir, os seus elementos, atributos, entidades e as súas posibles combinacións e anidamentos.

Os documentos XML ben formados que cumpren as restricións indicadas dize que son documentos validados ou válidos.

Estudaremos dous mecanismos para realizar dita validación: DTD e XML Schemas.

DTDs vs. XML Schemas

Os XML Schemas, máis modernos, teñen varias vantaxes con respecto ás DTDs:

- ‡ son, a súa vez, arquivos XML.
- ‡ pódense especificar os tipos de datos do contido, dos atributos, etc.
- ‡ son extensibles.

En posteriores epígrafes verase en detalle a validación dos documentos XML.

XML: Especificacións relacionadas co XML

Especificacións relacionadas co XML

Existen multitude de especificacións para traballar con documentos XML, moitas delas están aínda nunha fase inicial do seu desenvolvemento.

- ‡ Definición do tipo de documento: DTD, XML Schema.
- ‡ Enlazar entre partes ou documentos „ XML: XLink, XPointer.
- ‡ Buscar/filtrar información no documento: XPath.
- ‡ Creación de formularios/interfaces de usuario: XForms.
- ‡ Presentación e/ou transformación de documentos: CSS, XSL, XSLT.
- ‡ Linguaxe de consulta de bases de datos XML: XQuery.

XML: Ambitos de aplicación

...mbitos de aplicación

As vantaxes que achega o formato XML para almacenar e transportar información fai que se utilice en multitude de ámbitos, como por exemplo:

- ‡ Documentos ofimáticos: formato open document (ODF *f* OASIS Open Document Format for Office Applications), e documentos Microsoft Office (OOXML *f* Office Open XML)
- ‡ Protocolos estándar: WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol)
- ‡ Subscripción a fontes de información (feeds): RSS e ATOM
- ‡ Novos arquivos de configuración de aplicacións UNIX/LINUX
- ‡ Novas linguaxes para a web: XHTML *f* a nova versión de HTML creada en XML, SMIL (Synchronized Multimedia Integration Language) *f* a nova linguaxe (estándar aberto W3C ^[1]) para presentacións multimedia, WML (Wireless Markup Language) *f* linguaxe para páxinas de dispositivos móbiles con tecnoloxía WAP
- ‡ AJAX (Asynchronous JavaScript + XML)

Software para edición de XML

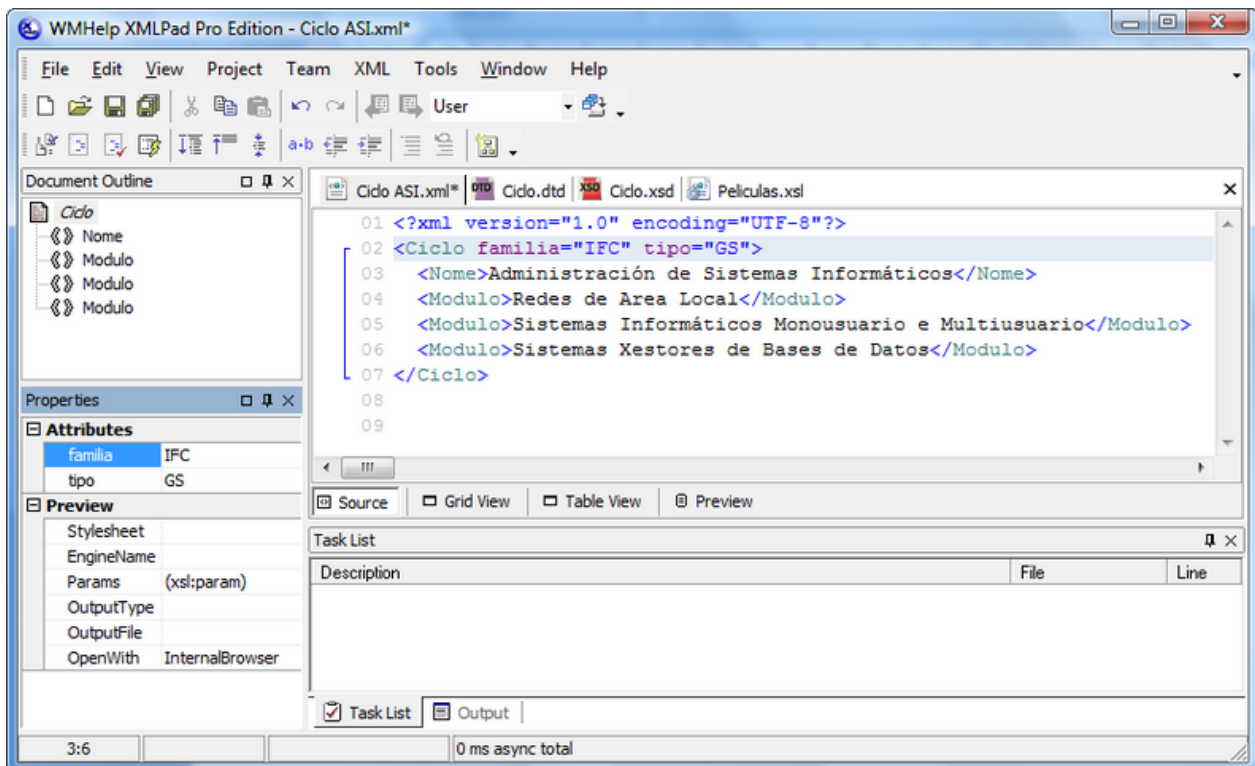
Aplicacións para edición de contido XML

Imos botar unha ollada ao panorama actual de aplicacións específicas para o manexo de documentos en XML. Comentaremos as características principais das aplicacións máis empregadas.

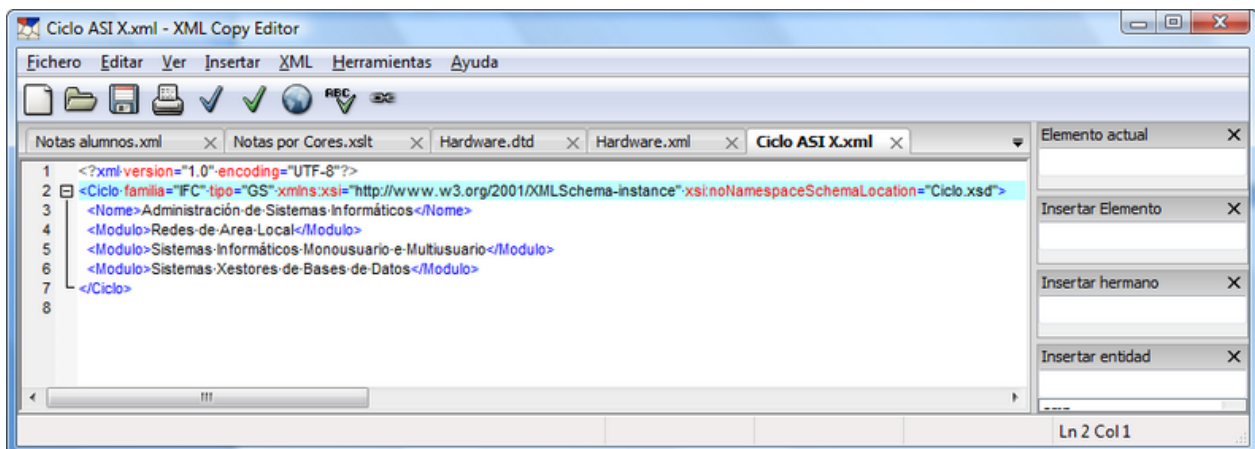
Aplicacións XML			
Nome	URL	Licenza	Sistemas
XMLPad	[1] [1]	Free	Win
XML Copy Editor	[3] [2]	Free	Win, Lin
Altova XML Spy	http://www.altova.com [4]	Com (Trial30)	Win
Liquid XML Studio	[5] [5]	Com (Trial30) / Comm. Ed.	Win
<oXygen/> XML Editor	http://www.oxygenxml.com [6]	Com (Trial30)	Win, Lin
Editix XML Editor	http://www.editix.com [7]	Free / Com (Trial30)	Win, Lin
CookTop	http://xmlcooktop.com [8]	Free	Win

Imos ver as capacidades de algunhas:

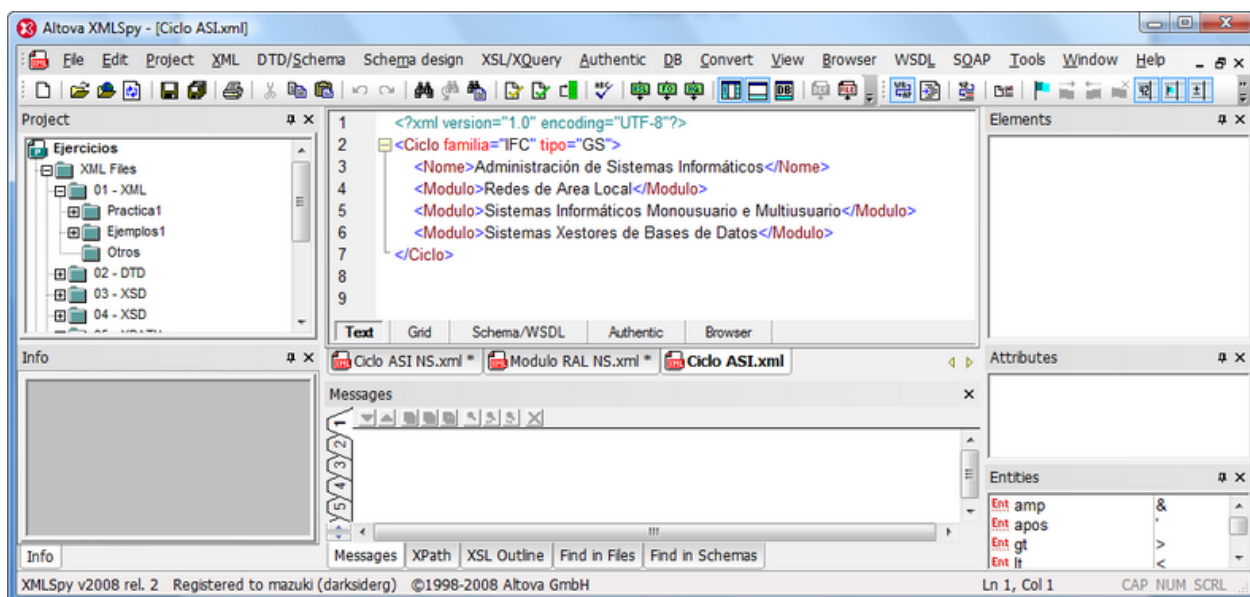
- ‡ **XMLPad**, a aplicación coa que recomendamos traballar ao longo deste curso. ^ de manexo sinxelo e inclúe todas as funcionalidades que necesitaremos para levar a cabo as actividades, entre outras:
 - ‡ Axuda na edición de documentos XML.
 - ‡ Comprobación automática de erros na composición dos documentos.
 - ‡ Xestión de proxectos.
 - ‡ Validación de documentos empregando DTDs e XML Schemas.
 - ‡ Avaliación de expresións XPath.
 - ‡ Aplicación de transformacións XSLT. Podedes consultar a continuación uns pequenos consellos para traballar con XSLT en XMLPad



‡ **XML Copy Editor**, unha ferramenta que tam, n incorpora entre as sfas caracter..sticas todo o necesario para realizar o curso. ^ un pouco m•is limitada que XMLPad, pero inclfe versi•ns para sistemas operativos Windows e Linux.



‡ **Altova XMLSpy**, a aplicaci•n m•is potente das presentes, con unha orientaci•n claramente profesional. S• ten versi•n comercial, e unicamente para sistemas Windows, a..nda que ofrecen unha licenza de proba v•lida durante 30 d..as.



Referencias

- [1] <http://www.wmhelp.com/xmlpad3.htm>
- [2] <http://xml-copy-editor.sourceforge.net/>
- [3] <http://xml-copy-editor.sourceforge.net>
- [4] <http://www.altova.com/>
- [5] <http://www.liquid-technologies.com/XmlStudio>
- [6] <http://www.oxygenxml.com/>
- [7] <http://www.editix.com/>
- [8] <http://xmlcooktop.com/>

Fonte dos artigos e contribuíntes

XML \$Fonte: <http://www.plategaxml.es/index.php?oldid=506> \$Contribuíntes: Victor

Linguaxes de marcas \$Fonte: <http://www.plategaxml.es/index.php?oldid=459> \$Contribuíntes: Victor

Codificaci3n de documentos \$Fonte: <http://www.plategaxml.es/index.php?oldid=475> \$Contribuíntes: Victor

XML: A linguaxe XML \$Fonte: <http://www.plategaxml.es/index.php?oldid=770> \$Contribuíntes: David, Victor

XML: O documento XML \$Fonte: <http://www.plategaxml.es/index.php?oldid=423> \$Contribuíntes: Victor

XML: Manipulaci3n de arquivos \$Fonte: <http://www.plategaxml.es/index.php?oldid=424> \$Contribuíntes: Victor

XML: Validaci3n de documentos \$Fonte: <http://www.plategaxml.es/index.php?oldid=425> \$Contribuíntes: Victor

XML: Especificaci3ns relacionadas co XML \$Fonte: <http://www.plategaxml.es/index.php?oldid=463> \$Contribuíntes: Victor

XML: Ambitos de aplicaci3n \$Fonte: <http://www.plategaxml.es/index.php?oldid=732> \$Contribuíntes: David, Victor

Software para edici3n de XML \$Fonte: <http://www.plategaxml.es/index.php?oldid=763> \$Contribuíntes: David, Victor

Fontes das imaxes, licenzas e contribuíntes

Image: Evolucion das linguaxes de marcas.jpg \$Fonte: http://www.plategaxml.es/index.php?title=Ficheiro:Evolucion_das_linguaxes_de_marcas.jpg \$Licenza: descoñecido \$Contribuíntes: Victor

Image: Opciones de codificación no gardado.png \$Fonte: http://www.plategaxml.es/index.php?title=Ficheiro:Opciones_de_codificacion_no_gardado.png \$Licenza: descoñecido \$Contribuíntes: Victor

Image: XMLPad.png \$Fonte: <http://www.plategaxml.es/index.php?title=Ficheiro:XMLPad.png> \$Licenza: descoñecido \$Contribuíntes: Victor

Image: XML Copy Editor.png \$Fonte: http://www.plategaxml.es/index.php?title=Ficheiro:XML_Copy_Editor.png \$Licenza: descoñecido \$Contribuíntes: Victor

Image: Altova XMLSpy.png \$Fonte: http://www.plategaxml.es/index.php?title=Ficheiro:Altova_XMLSpy.png \$Licenza: descoñecido \$Contribuíntes: Victor

Licenza

Attribution-Noncommercial-Share Alike 3.0 Unported
<http://creativecommons.org/licenses/by-nc-sa/3.0/>
