

UNIDADE 1

**RECOÑECEMENTO DAS CARACTERÍSTICAS
DE LINGUAXES DE MARCAS**

**LINGUAXES DE MARCAS E SISTEMAS
DE XESTIÓN DE INFORMACIÓN**

CICLO SUPERIOR DE ASIR



Índice

1 INTRODUCCIÓN.....	3
2 EVOLUCIÓN DAS LINGUAXES DE MARCAS.....	4
2.1 GML (Generalized Markup Language).....	6
2.2 SGML (Standard Generalized Markup Language).....	7
2.3 HTML (HyperText Markup Language).....	7
2.4 XML (eXtensible Markup Language).....	9
2.5 XML vs HTML.....	10
2.6 XML vs SGML.....	11
2.7 Etiquetas.....	11
2.8 Ferramentas de edición.....	11
3 XML: ESTRUCTURA E SINTAXE.....	12
3.1 O prólogo.....	14
3.2 Elementos. O raíz (exemplar) e o resto de elementos.....	15
3.2.1 Atributos.....	17
4 DOCUMENTOS BEN FORMADOS.....	18
5 USO DE ESPAZOS DE NOMES EN XML.....	19
5.1 Exemplo: uso de espazos de nomes.....	19



1 INTRODUCCIÓN

Unha "linguaxe de marcas" ofrece un mecanismo de codificar un documento onde, xunto co texto, se incorporan etiquetas, marcas ou anotacións con información adicional sobre a estrutura do propio texto ou do seu formato de presentación. Permiten facer explícita a estrutura dun documento, o seu contido semántico ou calquera outra información lingüística ou extra-lingüística que se desexa deixar clara.

Cada linguaxe de marcas defínese nun documento chamado DTD (Document Type Definition) no que se establecen as marcas permitidas, os elementos empregados por dita linguaxe, as súas correspondentes etiquetas e atributos, a súa sintaxe, e as normas de uso.

Exemplo

Aspecto dun documento feito nunha linguaxe de marcas:

```
<carta>  
  <data>12/09/2023</data>  
  <presentacion>Estimado cliente:</presentacion>  
  <contido>bla bla bla bla ...</contido>  
  <sinatura>Don Brais Souto García</sinatura>  
</carta>
```

As linguaxes de marcas confúndense a miúdo coas linguaxes de programación. Non obstante, non son iguais, xa que a linguaxe de marcas non ten funcións aritméticas ou variables, como teñen as linguaxes de programación.

As linguaxes de marcas pódense clasificar do seguinte xeito:

- **Presentación:** define o formato do texto.
- **Procedimental:** Orientado tamén á presentación pero, neste caso, o programa que representa o documento debe interpretar o código na mesma orde en que aparece.
- **Descritivos ou semánticos:** describen as distintas partes nas que se estrutura o documento pero sen especificar como deben representarse.

Na práctica, pódense combinar varias clases diferentes nun mesmo linguaxe de marcas.

Algúns exemplos de linguaxes de marcas agrupados polo seu ámbito de uso son:



- **Documentación electrónica:**

- **RTF (Rich Text Format):** [Formato de texto enriquecido](#), foi desenvolvido por Microsoft en 1987. Permite o intercambio de documentos de texto entre distintos procesadores de texto.
- **TeX:** O seu obxectivo é a creación de [ecuacións matemáticas complexas](#).
- **Wikitexto:** Permite a creación de páxinas wiki en servidores preparados para soportar este idioma.
- **DocBook:** Permítelle xerar documentos separando a estrutura lóxica do documento do seu formato. Deste xeito, estes documentos pódense publicar en diferentes formatos sen necesidade de realizar modificacións no documento orixinal.

- **Tecnoloxías de Internet:**

- **HTML, XHTML:** (Hypertext Markup Language, eXtensible Hypertext Markup Language): O seu obxectivo é a creación de páxinas web.
- **RSS:** Permite a difusión de contidos web.

- **Outras linguas especializadas:**

- **MathML (Mathematical Markup Language):** O seu obxectivo é expresar o formalismo matemático de forma que poida ser entendido por diferentes sistemas e aplicacións.
- **VoiceXML (Voice Extended Markup Language):** pretende intercambiar información entre un usuario e unha aplicación con capacidades de recoñecemento de voz.
- **MusicXML (Music Extended Markup Language):** Permite o intercambio de partituras entre diferentes editores de partituras.

2 EVOLUCIÓN DAS LINGUAXES DE MARCAS

Nos anos 70 xurdiron algunhas linguaxes informáticas, distintas das linguaxes de programación, orientadas á xestión da información. Co desenvolvemento dos editores e procesadores de texto apareceron as primeiras linguaxes informáticas especializadas en tarefas de descrición e estruturación da información: as linguaxes de marcas.



Ao mesmo tempo, xurdiron outras linguaxes informáticas, orientadas á representación, almacenamento e consulta de xeito eficiente de grandes cantidades de datos: linguaxes e sistemas de bases de datos.

As linguaxes de marcas xurdiron inicialmente como linguaxes formadas polo conxunto de códigos de formato que os procesadores de textos insiren nos documentos para dirixir o proceso de presentación (impresión), a través dunha impresora.

Como no caso das linguaxes de programación, inicialmente estes códigos de formato estaban ligados ás características dunha máquina, programa ou procesador de textos concreto e, nun principio, non había nada neles que permitise ao programador («formateador» de documentos neste caso) abstraerse das características do procesador de textos, e expresar de xeito independente a este, a súa estrutura e a lóxica interna do documento.

Exemplo

Código de marca comercial anterior a GML (as etiquetas son unha invención miña)

Dado o seguinte documento:

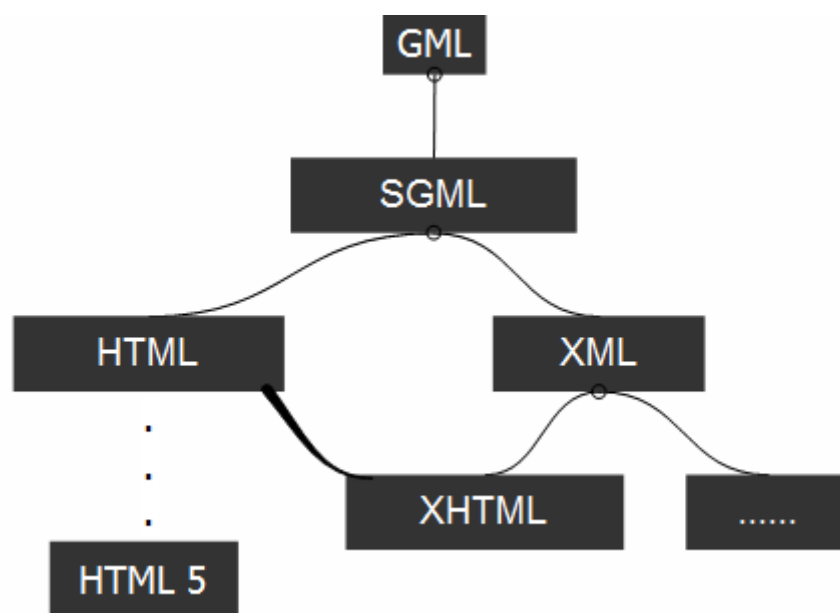
```
<times 14>
  <color verde>
    <center>
      Este texto é un exemplo para mostrar a utilización primitiva das marcas
    </center>
  </color>
</times 14>
<color granate>
  <times 10>
    <cursive>Para realizar este exemplo úsanse etiquetas da nosa invención. </cursive>
      As partes importantes do texto poden resaltarse usando a letra
      <bold>grosa</bold>, ou o <underline>suliñado</underline>
    </times 10>
  </color>
```

Máis tarde, este tipo de funcións engadíronse como medio de presentación á pantalla. Os códigos de estilo de visualización anteriores xa non aparecían e utilizáronse outros medios de marcado que non fosen incluír as cadeas de formato manualmente. Este proceso foi

automatizado e bastaba con premer unha combinación de teclas, ou un botón, para conseguir os resultados requiridos. Aínda que isto era só unha abstracción, para uso interno, as aplicacións aínda usaban marcas, para delimitar aquelas partes do texto que tiñan un formato especial.

Este marcado orientouse exclusivamente á presentación da información, aínda que pronto se decataron das posibilidades do marcado, e déronlle novos usos que resolvían unha gran variedade de necesidades. Deste xeito apareceu o formato xeneralizado.

Na seguinte imaxe amosamos un pequeno resumo da evolución das linguaxes de marcas que veremos ao longo desta unidade.



Imaxe: cadro coa evolución das linguaxes de marcado

2.1 GML (Generalized Markup Language)

Un dos problemas que apareceron hai décadas na informática é a falta de estandarización dos formatos de información empregados polos diferentes programas.

Para resolver este problema, na década de 1960 IBM encargou a Charles F. Goldfab a construción dun sistema legal de edición, almacenamento e busca de documentos. Despois de analizar o funcionamento da empresa, chegaron á conclusión de que para realizar un bo tratamento informático dos documentos había que establecer un formato normalizado para todos os documentos que se manexaban. Deste xeito foi posible xestionar calquera documento en calquera departamento e con calquera aplicación, sen ter en conta onde nin con que se xerou o documento. Dito formato debía ser válido para



os distintos tipos de documentos xurídicos empregados pola empresa, polo que debía ser flexible para poder axustarse ás diferentes situacións.

O formato de documento que se creou como resultado deste traballo foi GML, cuxo obxectivo era describir os documentos de forma que o resultado fose independente da plataforma e da aplicación utilizada.

2.2 SGML (Standard Generalized Markup Language)

O formato GML evolucionou ata 1986, cando deu lugar ao estándar ISO 8879, que se chamou SGML. Esta era unha linguaxe moi complexa e requiría ferramentas de software caras. Por este motivo, o seu uso quedou relegado a grandes aplicacións industriais.

Exemplo de documento SGML

```
<email>
  <remitente>
    <persoa>
      <nome> Xoel </nome>
      <apelido> Afonso </apelido>
    </persoa>
  </remitente>
  <destinatario>
    <enderezo> xoel@gmail.com </enderezo>
  </destinatario>
  <asunto>Quedamos hoxe?</asunto>
  <mensaxe> Ola, vin que esta noite botan a película que querías ver. Queres ir?</mensaxe>
</email>
```

2.3 HTML (HyperText Markup Language)

En 1989/90 Tim Berners-Lee creou a World Wide Web e atopouse coa necesidade de organizar, vincular e compatibilizar unha gran cantidade de información de diferentes sistemas. Para solucionar isto, creou unha linguaxe de descrición de documentos chamada HTML, que en realidade era unha combinación de dous estándares existentes:

- **ASCII:** é o formato que calquera procesador de textos pode recoñecer e almacenar. Polo tanto, é un formato que permite a transferencia de datos entre distintos ordenadores.
- **SGML:** Linguaxe que permite dar estrutura ao texto, destacando títulos ou



aplicando varios formatos ao texto.

HTML é unha versión simplificada de SGML, xa que só se utilizaron as instrucións absolutamente esenciais. Era tan fácil de entender que rapidamente gañou unha ampla aceptación, acadando o que SGML non puido.

HTML converteuse nun estándar xeral para crear páxinas web. Ademais, dende a súa creación, tanto as ferramentas de software como os navegadores, que permiten ver páxinas HTML, non deixaron de mellorar.

A pesar de todas estas vantaxes, HTML non é unha linguaxe perfecta, as súas principais desvantaxes son:

- Non admite tarefas de impresión e deseño.
- O idioma non é flexible, xa que as etiquetas son limitadas.
- Non permite mostrar contido dinámico.
- A estrutura e o deseño mestúranse no documento.

Exemplo de documento HTML:

```
<html>
  <head>
    <title> Exemplo de código HTML </title>
  </head>
  <body>
    <p></p>
    <p>
      <b>25 de xullo de 2023</b>
    </p>
    <p><b> Benvido ao modulo de "LMSXI" </b></p>
    <p> Neste curso aprender&aacute;s, entre outras cousas:<br/>
      <ul>
        <li>As ventaxes que ofrece XML </li>
        <li>A creaci&oacute;n de documentos ben formados </li>
        <li>A creaci&oacute;n de DTD</li>
      </ul>
    </p>
  </body>
</html>
```




2.4 XML (eXtensible Markup Language)

Para resolver os problemas de HTML, o [W3C](#) estableceu, en 1998, o estándar internacional XML, unha linguaxe de marcas puramente estrutural que **non inclúe ningunha información relacionada co deseño**. Estase a converter nun estándar para o intercambio de datos na web.

A diferenza do HTML, as etiquetas indican o significado dos datos en lugar do formato no que se mostrarán. XML é unha metalinguaxe caracterizada por:

- Permitir definir etiquetas propias.
- Permitir asignar atributos ás etiquetas.
- Usa un esquema para definir con precisión as etiquetas e os atributos permitidos.
- A estrutura e o deseño son independentes entre si.

En realidade, XML é un conxunto de estándares interrelacionados que son:

- **XSL:** Linguaxe de estilo extensible. Permítelle definir follas de estilo para documentos XML e inclúe capacidade para a transformación de documentos.
- **XML Linking Language:** Inclúe XPath, Xlink e Xpointer. Determinan aspectos sobre as ligazóns entre documentos XML.
- **XML Namespaces:** Proporcionan un contexto ao que se aplican as marcas dun documento XML e que serve para diferenciarlos doutros co mesmo nome válidos noutros contextos.
- **XML Schemas:** Permítenlle definir restricións que se aplicarán a un documento XML. Actualmente os máis utilizados son os DTD.

Exemplo de documento XML:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE biblioteca>
<biblioteca>
  <exemplar tipo_exem="libro" titulo="XML practico" editorial="Edicións Eni">
    <tipo> <libro isbn="978-2-7460-4958-1" edicion="1" paxinas="347"></libro> </tipo>
    <autor nome="Sebastien Lecomte"></autor>
    <autor nome="Thierry Boulanger"></autor>
    <autor nome="Ángel Belinchon Calleja" funcion="traductor"></autor>
    <prestado lector="Fiz Pardo">
```



```
<data_pres dia="13" mes="mar" ano="2023"></data_pres>
<data_devol dia="21" mes="xuñ" ano="2023"></data_devol>
</prestado>
</exemplar>
<exemplar tipo_exem="revista" titulo="Virtualización en GNU/Linux" editorial="Studio Press">
  <tipo>
    <revista>
      <data_publicacion mes="abr" ano="2023"></data_publicacion>
    </revista>
  </tipo>
  <autor nome="Varios"></autor>
  <prestado lector="Pedro Souto">
    <data_pres dia="12" mes="xan" ano="2023"></data_pres>
  </prestado>
</exemplar>
</biblioteca>
```

2.5 XML vs HTML

A continuación atoparás unha táboa comparativa das dúas linguaxes de marcas.

XML	HTML
É un perfil de SGML.	É unha aplicación de SGML .
Especifica como se deben definir os conxuntos de etiquetas aplicables a un tipo de documento.	Aplica un conxunto limitado de etiquetas a un único tipo de documento.
Modelo de hiperenlaces complexo.	Modelo de hiperenlaces simple.
O navegador é unha plataforma para desenvolver aplicacións.	O navegador é un visor de páxinas.
Fin da guerra dos navegadores e das etiquetas propietarias.	O problema da "non compatibilidade" e as diferenzas entre navegadores chegou a un punto no que a solución é difícil.



2.6 XML vs SGML

A continuación atoparás unha táboa comparativa das dúas linguaxes de marcas.

XML	SGML
Uso sinxelo	Uso complexo
Traballar con documentos ben formados. Non require validación.	Só funciona con documentos válidos.
Facilita o desenvolvemento de aplicacións de baixo custo.	A súa complexidade fai que as aplicacións informáticas para procesar SGML sexan moi caras
É moi utilizado en informática e noutras áreas de aplicación.	Só se emprega en sectores moi concretos.
Compatibilidade e integración con HTML	Non hai definida unha compatibilidade con HTML.
Formatos e estilos fáciles de aplicar	Formatos e estilos relativamente complexos
Non usa etiquetas opcionais	

2.7 Etiquetas

As linguaxes de marcado usan unha serie de etiquetas especiais intercaladas nun documento de texto sinxelo. Estas etiquetas serán interpretadas posteriormente por intérpretes lingüísticos e axudarán a procesar o documento.

As etiquetas escríbense encerradas entre ángulos, é dicir, < e >. Normalmente utilízanse dúas etiquetas: un inicio e un final para indicar que o efecto que queriamos presentar rematou. A única diferenza entre os dous é que o de peche ten unha barra inclinada "/" antes do código.

<tag>texto que sufrirá as consecuencias da etiqueta</tag>

<u>Esto está **suliñado**</u>

As últimas especificacións emitidas polo [W3C](#) sinalan a necesidade de que se escriban sempre en minúsculas para considerar que o documento está correctamente creado.



2.8 Ferramentas de edición

Para traballar en XML é necesario editar os documentos e procesalos, polo que temos dous tipos de ferramentas:

- **Editores XML:** Unha característica das linguaxes de marcado é que se basean no uso de ficheiros de texto sinxelos, polo que é suficiente empregar un procesador de textos normal para construír un documento XML.

Para crear documentos XML complexos e recoller datos, é recomendable utilizar un editor XML. Estes axúdannos a crear estruturas e etiquetas, dous elementos empregados nos documentos, e algúns tamén inclúen axuda para a creación doutros elementos como formatos de estilo DTD, CSS ou XSL,... Ou o W3C desenvolve un editor HTML, XHTML, Free. CSS e XML cuxo nome é Amaya.

- **Procesadores XML:** Os procesadores XML permítenche ler documentos XML e acceder ao seu contido e estrutura.
 - Un procesador é un conxunto de módulos de software, entre os que se atopa un parser ou analizador XML, que comproba que o documento cumpre os estándares establecidos para poder ser aberto.
 - Os procesadores XML poden obrigarche a traballar só con documentos dun tipo válido (entón chamados "validadores") ou poden só exixir que o documento estea ben formado ("non validadores").
 - O xeito no que os procesadores ler os datos XML descríbese na recomendación XML establecida polo W3C.
 - Para publicar un documento XML en Internet utilízanse procesadores XSLT, que permiten extraer ficheiros HTML dos documentos XML.
 - Para interpretar o código XML pódese usar calquera navegador.
 - XML tamén se pode usar para o intercambio de datos entre aplicacións. Neste caso, temos que recorrer a motores independentes, que funcionan sempre que teñamos coñecemento deles. Por exemplo, [JAXP de Oracle](#).

3 XML: ESTRUCTURA E SINTAXE

O XML, ou Extended Label Language, é unha linguaxe de etiquetas, creada polo programador, que estrutura e almacena información de forma ordenada. Non representa datos por si só, só organiza a estrutura.



XML aforra tempo de desenvolvemento e ofrece vantaxes, proporcionando aos sitios web e ás aplicacións unha forma realmente poderosa de almacenar información. Ademais, converteuse nun formato universal que foi asimilado por todo tipo de sistemas operativos e dispositivos móbiles.

Como en HTML, un documento XML é un documento de texto, neste caso cunha extensión ".xml", formado por pares de etiquetas, estruturadas nunha árbore, que describen unha función na organización do documento, que pode ser editada. con calquera editor de texto e que é interpretado polos navegadores web.

As características básicas de XML son:

- Como XML foi deseñado para funcionar na web, é directamente compatible con protocolos que xa funcionan, como HTTP e URL.
- Calquera documento que comprobe as regras de XML é compatible con SGML.
- Non se requiren coñecementos de programación para realizar tarefas sinxelas en XML.
- Os documentos XML son fáciles de crear.
- A difusión de documentos XML está asegurada xa que calquera procesador XML pode ler un documento XML.
- O marcado XML é lexible por humanos.
- O deseño XML é formal e conciso.
- XML é extensible, adaptable e aplicable a unha gran variedade de situacións.
- XML está orientado a obxectos.
- Cada documento XML está formado exclusivamente por datos de marcado e datos de caracteres mesturados.

O proceso de creación dun documento XML pasa por varias etapas nas que o éxito de cada unha delas baséase na calidade do anterior. Estas etapas son:

- Especificación de requisitos.
- Deseño de etiquetas.
- Marcación de documentos.

As marcas XML son etiquetas que se engaden ao texto para estruturar o contido do



documento. Esta información adicional permite aos ordenadores "interpretar" os textos. O marcado é todo o que hai entre os caracteres "<" e ">" ou "&" e ";"

Os datos de carácter son os que forman a verdadeira información do documento XML.

O marcado pode ser tan rico como se queira. Pode ser interesante detectar necesidades futuras e crear documentos cunha estrutura facilmente actualizables.

Os documentos XML poden ter comentarios, que non son interpretados polo intérprete XML. Inclúense entre as cadeas "<!--" e "-->", poden estar en calquera posición do documento excepto:

- Antes do prólogo.
- Dentro dunha etiqueta.

Os documentos XML poden estar formados por unha parte opcional chamada prólogo e outra parte obrigatoria denominada exemplar.

3.1 O prólogo

Se un documento XML inclúe prólogo, este debe preceder ao exemplar do documento. A súa inclusión facilita o tratamento da información do exemplar. O prólogo divídese en dúas partes:

- **A declaración XML:** Se está incluída, debe ser a primeira liña do documento, se non, xérase un erro que impide que o documento sexa procesado.

O feito de ser opcional permite procesar documentos HTML e SGML coma se fosen XML; se fose obrigatorio, terían que incluír unha declaración de versión XML, que non teñen.

O prólogo pode ter tres funcións:

- Declaración da versión de XML utilizada para preparar o documento: Para iso utilízase a etiqueta:

```
<?xml versión= "1.0" ?>
```

Neste caso indica que o documento foi creado para a versión XML 1.0.

- Declaración da codificación utilizada para representar os caracteres: Determina o conxunto de caracteres utilizado no documento. Para iso escribes:



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

Neste caso utilízase o código iso-8859-1 (Latín-1), que permite o uso de acentos ou caracteres como ñ. Os códigos máis importantes son:

ESTÁNDAR ISO	CÓDIGO DO PAÍS
UTF-8 (Unicode)	Conjunto de caracteres universal
ISO-8859-1 (Latin-1)	Europa occidental, Latinoamérica
ISO-8859-2 (Latin-2)	Europa central y oriental
ISO-8859-3 (Latin-3)	Sudoeste de Europa
ISO-8859-4 (Latin-4)	Países Escandinavos, Bálticos
ISO-8859-5	Cirílico
ISO-8859-6	Árabe
ISO-8859-7	Griego
ISO-8859-8	Hebreo
ISO-8859-9	Turco
ISO-8859-10	Lapón. Nórdico, esquimal
EUC-JP oder Shift_JIS	Japonés

- Declaración de autonomía do documento: Informa se o documento precisa doutro para a súa interpretación. Para declaralo, tes que definir todo o prólogo:

```
<?xml version= "1.0" encoding="iso-8859-1" standalone="yes" ?>
```

Neste caso, o documento é independente, se non, o atributo autónomo tomaría o valor "non".

- **A declaración do tipo de documento**: define que tipo de documento estamos a crear para que se procese correctamente. Cada declaración de tipo de documento comeza coa cadea:

```
<!DOCTYPE Nome_tipo ...>
```



3.2 Elementos. O raíz (exemplar) e o resto de elementos.

É a parte máis importante dun documento XML, xa que contén os datos reais do documento. Está formado por elementos aniñados.

Os elementos son os diferentes bloques de información que definen a estrutura dun documento XML. Están delimitados por unha etiqueta de apertura e unha etiqueta de peche. Á súa vez, os elementos poden estar formados por outros elementos e/ou atributos.

Exemplo:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE libro>
<libro>
  <titulo>XML practico </titulo>
  <autor>Sebastien Lecomte</autor>
  <autor>Thierry Boulanger</autor>
  <editorial>Ediciones Eni</editorial>
  <isbn>978-2-7460-4958-1</isbn>
  <edicion>1</edicion>
  <paxinas>347</paxinas>
</libro>
```

O exemplar é o elemento `<libro>`, que a súa vez está composto dos elementos `<autor>`, `<editorial>`, `<isbn>`, `<edicion>` e `<paxinas>`

En realidade, a instancia é o elemento raíz dun documento XML. Todos os datos dun documento XML deben pertencer a un elemento do mesmo.

Os nomes das etiquetas deben ser autodescritivos, o que facilita o traballo realizado con elas.

A formación dos elementos debe cumprir uns estándares para que estean perfectamente definidos e que o documento XML ao que pertencen poida ser interpretado polos procesadores XML sen xerar erros mortais. Estas regras son:

- En cada documento XML debe haber un elemento raíz e só un.
- Todos os elementos teñen unha etiqueta de inicio e unha etiqueta de final. No caso de que haxa elementos baleiros no documento, as etiquetas de inicio e final pódense substituír por unha etiqueta de elemento baleira. Constrúese como a



etiqueta de inicio, pero substituíndo o carácter ">" por ">". É dicir, <elemento></elemento> pódese substituír por: <elemento/>

- Ao aniñar elementos, teña en conta que non se pode pechar un elemento que conteña outro elemento que aínda non se pechou.
- Os nomes das etiquetas de inicio e final dun mesmo elemento deben ser idénticos, respectando maiúsculas e minúsculas. Poden ser calquera cadea alfanumérica que non conteña espazos e non comece polo carácter de dous puntos, ":", nin coa cadea "xml" ou calquera das súas versións nas que se cambien as letras maiúsculas e minúsculas ("XML", "XmL", "xML",...).
- O contido do elemento non pode conter a cadea "]]>" para compatibilidade con SGML. Ademais, os caracteres maiores que, >, menores que, <, ampersand, &, comiñas dobres, ", e apóstrofo, ' non se poden usar directamente. Se hai que usar estes caracteres, substitúense polas seguintes cadeas:

CARÁCTER	CADEA
>	>
<	<
&	&
"	"
'	'

Para usar caracteres especiais, como £, ©, ®,... debes usar as expresións &#D; ou &#H; onde D e H corresponden respectivamente ao número decimal ou hexadecimal correspondente ao carácter que se quere representar no código UNICODE. Por exemplo, para incluír o carácter euro, €, usaríamos € ou €

No seguinte [enlace](#) atoparás unha táboa cos caracteres ASCII, o nome HTML e o número HTML de cada un deles, que serán imprescindibles á hora de crear documentos HTML e XML.

3.2.1 Atributos

Permiten engadir propiedades aos elementos dun documento. Os atributos non se poden



organizar en ningunha xerarquía, non poden conter ningún outro elemento ou atributo e non reflicten ningunha estrutura lóxica.

Non se debe usar un atributo para conter información susceptible de ser dividida.

Exemplo:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE biblioteca >
<biblioteca>
  <exemplar tipo_exem="libro" titulo="XML práctico" editorial="Ediciones Eni">
    <tipo> <libro isbn="978-2-7460-4958-1" edicion="1" paxinas="347"></libro> </tipo>
    <autor nome="Sebastien Lecomte"></autor>
    <autor nome="Thierry Boulanger"></autor>
    <autor nome="Angel Belinchon Calleja" funcion="traductor"></autor>
    <prestado lector="Xoel Castelo">
      <data_pres dia="13" mes="mar" ano="2009"></data_pres>
      <data_devol dia="21" mes="jun" ano="2009"></data_devol>
    </prestado>
  </exemplar>
</biblioteca>
```

Como se ve no exemplo, os atributos defínense e dáselles un valor dentro dunha etiqueta de inicio ou elemento baleira, seguindo o nome do elemento ou a definición doutro atributo, sempre separados deles por un espazo. Os valores dos atributos van precedidos por un igual que segue ao nome do atributo e deben definirse entre comiñas simples ou dobres.

Os nomes de atributos deben seguir as mesmas regras que os nomes de elementos e non poden conter o carácter inferior a <.

4 DOCUMENTOS BEN FORMADOS

Todos os documentos XML deben verificar as regras de sintaxe definidas pola recomendación do W3C para o estándar XML. Esas regras básicas son:

- O documento debe ter un prólogo definido coa declaración xml completa.
- Hai un único elemento raíz para cada documento: é un único elemento no que están aniñados todos os demais elementos e contido.
- Deben seguirse as regras sintácticas da linguaxe XML para definir os distintos



elementos e atributos do documento.

5 USO DE ESPAZOS DE NOMES EN XML

Permiten definir a pertenza dos elementos e atributos dun documento XML ao contexto dun vocabulario XML. Resólvese así as ambigüidades que poden xurdir ao xuntar dous documentos diferentes, de dous autores diferentes, que utilizaron o mesmo nome de etiqueta para representar cousas distintas.

Os espazos de nomes, tamén coñecidos como "espazos de nomes", permítenche dar un nome único a cada elemento, indexándoos segundo o nome de vocabulario axeitado. Tamén están asociados cun URI que os identifica de forma única.

No documento, as etiquetas ambiguas substitúense por etiquetas nas que o nome do elemento vai precedido dun prefixo, que determina o contexto ao que pertence a etiqueta, seguido de dous puntos, ":". Isto é:

```
<prefixo:nome_etiqueta></prefixo:nome_etiqueta>
```

Esta etiqueta chámase "nome cualificado". Á hora de definir o prefixo hai que ter en conta que non se poden utilizar espazos ou caracteres especiais e que non comeza por un dígito.

Antes de poder usar un prefixo de espazo de nomes, para resolver a ambigüidade de dúas ou máis etiquetas, debes declarar o espazo de nomes, é dicir, asociar un índice co URI asignado ao espazo de nomes, mediante un atributo especial denominado **xm:ns**. Isto faise entre o prólogo e a instancia dun documento XML e a súa sintaxe é a seguinte:

```
<conexión>://<endereço_servidor>/<sección1>/<sección2>/...
```

5.1 Exemplo: uso de espazos de nomes

Supoñamos dous documentos que organizan a información sobre o profesorado e o alumnado do Ciclo Formativo.

- **XML de alumnos:**

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"  
<!DOCTYPE alumnos>  
<alumnos>
```



```
<nome>Uxía Fernández Igrexas</nome>  
<nome>Senén González Louzao</nome>  
<nome>Ricardo Lis López</nome>  
</alumnos>
```

- **XML de profesores:**

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>  
<!DOCTYPE profesores>  
<profesores>  
  <nome>Pilar Sueiro Pérez</nome>  
  <nome>Tomé Facal Conde</nome>  
</profesores>
```

Se uniramos os dous documentos nun só, sen utilizar espazos de nomes, non se distinguirían os profesores dos alumnos xa que en ambos casos a etiqueta <nome> leva o mesmo nome.

Para solucionar isto necesitamos definir un espazo de nomes para cada contexto.

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>  
<!DOCTYPE membros>  
<alumnos xmlns:alumnos="http://DAM/alumnos">  
<profesores xmlns:profesores="http://DAM/profesores">  
<asistentes>  
  <alumnos:nome>Fernando Fernández González</alumnos:nome>  
  <alumnos:nome>Isabel González Fernández</alumnos:nome>  
  <alumnos:nome>Ricardo Martínez López</alumnos:nome>  
  <profesores:nome>Pilar Ruiz Pérez</profesores:nome>  
  <profesores:nome>Tomás Rodríguez Hernández</profesores:nome>  
</asistentes>
```