



<b>1.</b>	<b>Introdución a linguaxe XML.....</b>	<b>3</b>
1.1	Vantaxes do emprego de XML.....	4
<b>2.</b>	<b>Contido dun documento XML .....</b>	<b>4</b>
2.1	Elementos.....	4
2.2	Nomes dos elementos .....	5
2.3	Atributos .....	6
2.4	Atributos ou subelementos .....	6
2.5	Valores dos atributos .....	6
2.6	Texto 7	
2.7	Comentarios .....	7
2.8	Instrucións de procesamento .....	8
<b>3.</b>	<b>Documentos XML ben formados .....</b>	<b>8</b>
3.1	Especificacións de XML.....	9
3.2	O prólogo.....	9
3.3	Documentos ben formados e navegadores web .....	10
3.4	Espazos en branco .....	11
3.5	Identificación da linguaxe do contido .....	12
3.6	As entidades.....	12
3.7	A sección CDATA.....	13
<b>4.</b>	<b>Necesidade dos espazos de nomes .....</b>	<b>13</b>
4.1	Declaración dun espazo de nomes.....	14
4.2	Espazos de nomes con prefixo .....	16
4.3	Ámbito dun espazo de nomes.....	17
4.4	Atributos e espazos de nomes.....	18

# 1. Introducción a linguaxe XML

---

A característica principal dunha linguaxe de marcas é que engade ao contido do texto marcas ou etiquetas que permiten engadir ao mesmo información adicional sobre a súa estrutura ou presentación.

HTML é sen dúbida a linguaxe de marcas máis empregada. Un documento sinxelo que empregue a linguaxe HTML podería ser por exemplo o seguinte:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Exemplo de HTML</TITLE>
  </HEAD>
  <BODY>
    <!-- Exemplo.html -->
    <H1>HTML 1.0</H1>
    <P>A primeira versión de HTML foi desenvolvida por Tim Berners-Lee.</P>
  </BODY>
</HTML>
```

Na linguaxe HTML existe un número de etiquetas predefinidas (HTML, HEAD, BODY, H1, P, ...). Nas primeiras versións de HTML empregábanse etiquetas para definir tanto a estrutura do documento como a súa aparencia, pero iso foi cambiando co tempo de xeito que na actualidade aconséllase definir a aparencia empregando follas de estilos CSS e deixar as etiquetas para definir soamente a estrutura das páxinas web.

A principal diferenza da linguaxe XML con outras linguaxes de marcas como HTML é que *en XML non existen etiquetas predefinidas*. O conxunto de etiquetas a empregar defínese para cada documento, de xeito que permitan identificar o seu contido. Por exemplo, un documento XML con información sobre persoas podería conter:

```
<apelido>Louzán</apelido>
<nome>Xaime</nome>
<datanacemento>04/08/1969</datanacemento>
```

E un documento XML sobre módulos profesionais:

```
<módulos>
  <módulo>Fundamentos hardware
    <sesións>4</sesións>
  </módulo>
  <módulo>Linguaxes de marcas e sistemas de información
    <sesións>5</sesións>
  </módulo>
</módulos>
```

Isto é, XML (*eXtensible Markup Language*, Linguaxe de Marcado Extensible) non é unha linguaxe de marcas. *XML é unha meta-linguaxe* coa que podemos definir as nosas propias linguaxes de marcas para cada necesidade específica. Empregando XML podemos crear linguaxes que almacenan a información que necesita unha aplicación concreta.

XML é un subconxunto da linguaxe SGML (*Standard Generalized Markup Language*, Linguaxe de Marcado Estándar Xeneralizado) 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.

## 1.1 Vantaxes do emprego de XML

A linguaxe XML creouse para estruturar, almacenar e transportar información. Os requisitos principais que se tiveron en conta na creación de XML foron:

- Que fose doado crear documentos XML e que se puideran ler e entender de forma sinxela.
- Que os documentos XML puideran ser procesados informáticamente de forma sinxela.
- Que puidera servir en múltiples campos mantendo ao mesmo tempo a compatibilidade con SGML.

As vantaxes principais derivadas do emprego de documentos XML son:

- A estrutura dun documento XML pódese entender sen dificultade.
- Aporta estrutura e información aos datos dos documentos; empregando correctamente os elementos da linguaxe podemos identificar as partes dun documento e o seu significado concreto.
- Facilita a comunicación de información entre aplicacións, de xeito independente da plataforma na que estean desenvolvidas e dos orixes de datos que empreguen.
- Os documentos XML pódense estender de forma sinxela; ao crear unha linguaxe baseada en XML garántese que nun futuro poderanse engadir novas etiquetas á mesma mantendo a compatibilidade coas xa existentes.
- Existen analizadores estándar de documentos XML que podemos empregar para procesar calquera linguaxe baseada en XML. Non é preciso crear analizadores específicos para cada linguaxe, o que acelera o desenvolvemento de aplicacións que utilicen documentos XML.

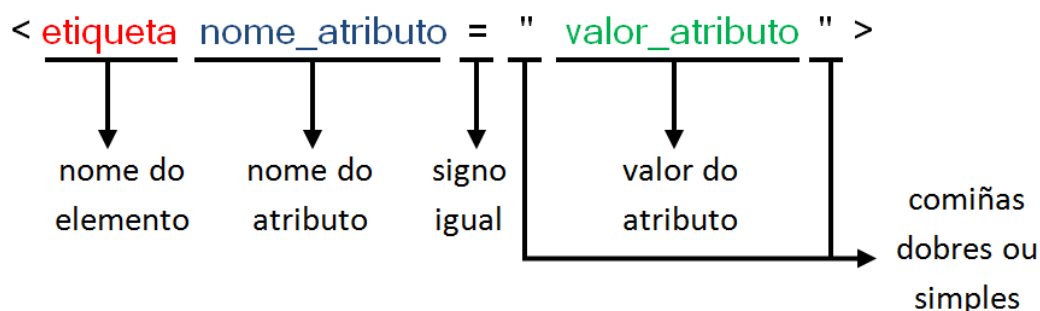


Antes de comezar a editar documentos empregando XML, debemos preparar un entorno adecuado. No TA1 explícanse algunhas das opcións que podemos utilizar para crear documentos na linguaxe XML.

## 2. Contido dun documento XML

### 2.1 Elementos

A base dun documento XML é o *elemento*. Cada elemento está formado por unha etiqueta de apertura e outra de peche, empregando en ambos casos os caracteres "<" e ">". O contido dun elemento é todo o que figura entre ámbalas dúas etiquetas.



Un elemento **pode conter:**

- **atributos**, que figurarán sempre na etiqueta de apertura.
- **texto**, entre a etiqueta de apertura e a etiqueta de peche.
- **outros elementos**.

Por exemplo, o elemento "módulo" contén os atributos "sesións" e "horas", un texto, e o subelemento "profesor":

```
<módulo sesións="5" horas="133">
    Linguaxes de marcas e sistemas de información
    <profesor>Xaime Louzán</profesor>
</módulo>
```

Tamén poden existir elementos baleiros, que poden ser representados por dúas ou por unha única etiqueta:

```
<activo></activo>
<activo />
```

## 2.2 Nomes dos elementos

Considérase como nome dun elemento o que figura na etiqueta de apertura e de peche, que deben coincidir.

O nome dun elemento pode conter calquera carácter alfabético (acentuado ou non), números e algúns signos de puntuación como ".", "\_" e "-", pero **non poden conter espazos**. Ademais, **o primeiro carácter tampouco pode ser un número nin algún dos caracteres "." e "-"**. Por exemplo, os seguintes nomes de elementos non serían válidos:

```
<1Enero2013></1Enero2010>
<+elemento></+elemento>
< .elemento></ .elemento>
<-elemento></-elemento>
```

Aínda que, segundo se reflexa na especificación da linguaxe, tamén se pode empregar o carácter ":" **nos nomes dos elementos, débese evitar** pois como veremos máis adiante leva a confusión coa sintaxe empregada nos espazos de nomes.

**Tamén están reservados os nomes de elementos que empezan polos caracteres "xml"**, en calquera combinación posible de minúsculas e maiúsculas, como "Xm1", "XmL" ou "xmL".

## 2.3 Atributos

Os elementos poden conter ou non atributos. Os atributos son un xeito de incorporar características ou propiedades aos elementos. Por exemplo:

```
<módulo sesións="5" horas="133"></módulo>
```

Cando un elemento contén atributos, deben figurar na etiqueta de apertura separados uns de outros e do nome do elemento por espazos.

Non existe un límite ao número de atributos máximo que pode conter un elemento, e tampouco ten importancia a orde na que estes figuren dentro do elemento, pero **un elemento concreto non pode ter atributos co nome repetido.**

**Os nomes dos atributos deben respectar as mesmas regras que os nomes dos elementos, en canto aos tipos de caracteres permitidos.**

## 2.4 Atributos ou subelementos

Ás veces temos que escoller entre empregar un subelemento ou un atributo para almacenar certa información. Por exemplo, para facer referencia ao profesor que imparte un módulo poderíamos poñer:

```
<módulo>Linguaxes de marcas e sistemas de información  
  <profesor>Xaime Louzán</profesor>  
</módulo>
```

Ou tamén:

```
<módulo profesor="Xaime Louzán">Linguaxes de marcas e sistemas de  
información</módulo>
```

Para facer a elección, temos que recordar que os atributos están pensados para gardar información (descricións) acerca dos datos e só poden almacenar un único valor, mentres que os elementos poden almacenar varios datos e incluso estenderse mediante estruturas aninhadas de varios elementos.

No exemplo anterior, soamente coa primeira solución poderíamos engadir información sobre o profesor, ou indicar un segundo profesor para ese mesmo módulo.

## 2.5 Valores dos atributos

O valor do atributo debe ir encerrado entre comiñas, simples ou dobres e sepárase do nome de atributo por un signo "=". **Algúns atributos poden empregar comiñas simples e outros atributos poden empregar comiñas dobres, pero non se poden mesturar** ambos tipos no mesmo atributo. Por exemplo, o seguinte atributo "atrib3" non sería válido:

```
<elemento atrib1="valor1" atrib2='valor2' atrib3="valor3"></elemento>
```

**En XML tódolos atributos dun elemento deben ter un valor definido. Non podemos ter atributos nos que non se indique o seu valor.** Por exemplo non sería válido:

```
<elemento atrib1></elemento>
```

No canto, teríamos que facer algo como:

```
<elemento atribl="sí"></elemento>
```

Ou tamén:

```
<elemento atribl=""></elemento>
```



Na tarefa 1 traballaremos cos elementos dun documento XML e os seus atributos.

## 2.6 Texto

Como parte do contido dun elemento, entre as etiquetas de apertura e peche do mesmo pode figurar texto, outros elementos, ou ámbolos dous ao mesmo tempo.

O texto dos documentos XML pode conter calquera tipo de carácter salvo "<" e "&". Así, non sería válido:

```
<autor>Stevie Ray Vaughan & Double Trouble</autor>
```

A mesma restrición aplícase aos valores dos atributos, que ademais dos caracteres "<" e "&" tampouco poden conter o mesmo tipo de comiña que se emprega para delimitalo. Por exemplo, non sería válido:

```
<canción nome='Don't bang the drum'></canción>
```

Se queremos empregar un destes caracteres dentro do texto ou dentro do valor dun atributo, temos dúas opcións:

- empregar unha *referencia a un carácter Unicode*. As referencias sempre empezan por "&#" (ou por "&#x" se empregamos hexadecimal) e rematan en punto e coma (;). Por exemplo, os caracteres "&" e "<" poderían reempazarse respectivamente polas referencias "&#38;" e "&#60;" (en hexadecimal serían "&#x26;" e "&#x3C;" respectivamente).
- empregar *entidades*. Como veremos máis adiante, empregando entidades poderíamos reempazar os caracteres "&" e "<" respectivamente polas cadeas "&amp;" e "&lt;"

Así, o primeiro exemplo anterior poderíamos poñelo como:

```
<autor>Stevie Ray Vaughan &#38; Double Trouble</autor>
```

Ou tamén:

```
<autor>Stevie Ray Vaughan &amp; Double Trouble</autor>
```

## 2.7 Comentarios

As veces é conveniente inserir comentarios nun documento XML. Estes serán ignorados cando se procese a información do documento. Teñen o mesmo formato que na linguaxe HTML, isto é, comezan pola cadea "<!--" e rematan con "-->".

```
<!-- Isto é un comentario -->
```

Poden aparecer en calquera lugar do documento onde poda ir texto, e poden conter calquera carácter excepto a combinación dobre guión "--".

## 2.8 Instrucións de procesamento

As instrucións de procesamento empréganse para dar información ás aplicacións que procesen os documentos XML. Aínda que poden aparecer en calquera parte dun documento, adoitan poñerse ao comezo do mesmo. Delimítanse polos conxuntos de caracteres "<?" e "?>". Por exemplo:

```
<?xml-stylesheet type="text/xsl" href="plantilla.xsl"?>
```

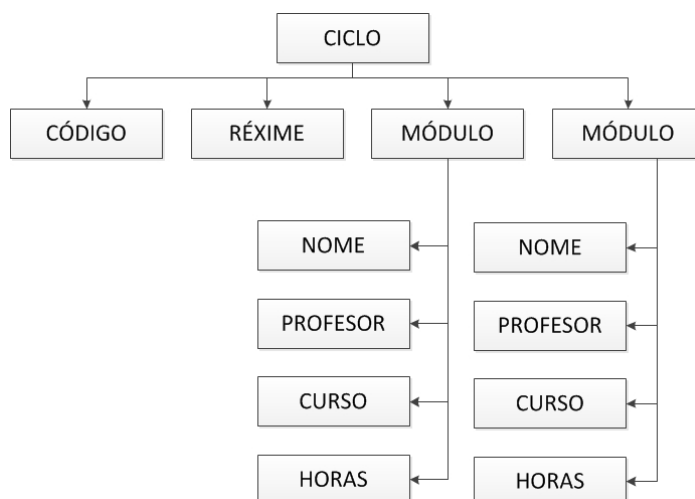
Traballaremos con instrucións de procesamento cando vexamos a linguaxe XSLT.

## 3. Documentos XML ben formados

Un documento considérase "*ben formado*" se respecta as normas que vimos ata o de agora, e tamén as seguintes descritas na especificación oficial de XML:

- Tódolos elementos do documento aníñanse en forma de árbore, existindo un único elemento raíz, que contén ao resto de elementos. Por exemplo:

```
<ciclo>
  <código>ASIR</código>
  <réxime>Ordinario</réxime>
  <módulo>
    <nome>Linguaxes de marcas e sistemas de información</nome>
    <profesor>Xaime Louzán</profesor>
    <curso>1º</curso>
    <horas>133</horas>
  </módulo>
  <módulo>
    <nome>Servizos de Rede e Internet</nome>
    <profesor>Ana Quintela</profesor>
    <curso>2º</curso>
    <horas>140</horas>
  </módulo>
</ciclo>
```





- **Tódolos elementos deben pecharse**, e isto debe facerse sempre de forma estruturada, é dicir, **en orden inversa a súa apertura**: o último elemento en abrirse será o primeiro en pecharse.

```
<elementoA>
  <elementoB>
    </elementoB>
  </elementoA>
```

- **A etiqueta de peche dun elemento debe coincidir exactamente co nome que figura na etiqueta de apertura**, incluíndo minúsculas e maiúsculas.
- Na especificación 1.1 da linguaxe, os documentos XML deben comezar cun prólogo.

## 3.1 Especificacións de XML

Existen dúas especificacións da linguaxe XML: a 1.0 e a 1.1. As diferenzas entre elas son escasas. A versión máis empregada hoxe en día é a 1.1, que engade principalmente sobre a anterior:

- Soporte Unicode completo. A especificación 1.0 límtase aos caracteres recollidos en Unicode 2.0, mentres que a especificación 1.1 abrangue incluso caracteres que se recollan en versións futuras de Unicode.
- A capacidade de recoñecer outros caracteres de fin de liña. En XML 1.0 podíamos empregar "&#xA;" para o salto de liña e "&#xD;" para o retorno de carro. A versión 1.1 engade soporte para os caracteres de fin de liña definidos por Unicode ("&#x2028;" e por EBCDIC ("&#x85;").
- A obrigatoriedade de engadir un prólogo aos documentos para que estean ben formados.

## 3.2 O prólogo

O prólogo dun documento XML debe aparecer **ao comezo** do mesmo, e **indicar obrigatoriamente** (en XML 1.1) a versión da linguaxe empregada. Ten a forma:

```
<?xml version="1.1"?>
```

Opcionalmente pode incluír tamén outros atributos:

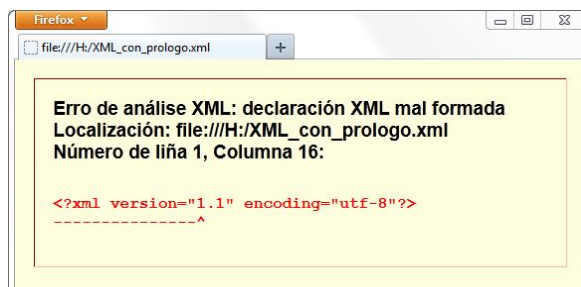
- **encoding**, para indicar o tipo de codificación que se emprega no documento.
- **standalone**, que como veremos na seguinte unidade utilízase para indicar se o documento depende ou non doutros documentos externos para comprobar a validez da gramática empregada no mesmo.

Tódolos procesadores XML deben ser capaces de ler documentos codificados en UTF-8 e UTF-16 e inferir por si mesmo cal de ambos é o empregado nun documento particular; se este é o caso non é preciso pero si aconsellable indicar o atributo encoding. Para outros sistemas de codificación (como os ISO-8859-x), débese comprobar que sexa admitido polo procesador (os máis habituais non presentaran problemas) e será obrigatorio especificar o atributo encoding no prólogo.

Por exemplo, un prólogo común dun documento XML é:

```
<?xml version="1.1" encoding="utf-8"?>
```

Debemos ter en conta que algúns navegadores non procesan correctamente os documentos XML que conteñan un prólogo. Por exemplo, IE9 e Firefox 18 non amosan os documentos cando a versión de XML que se indica no prólogo é a 1.1.

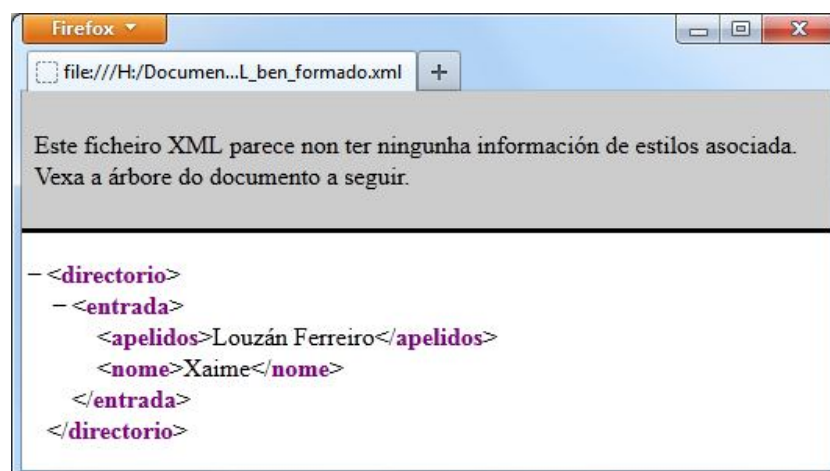


### 3.3 Documentos ben formados e navegadores web

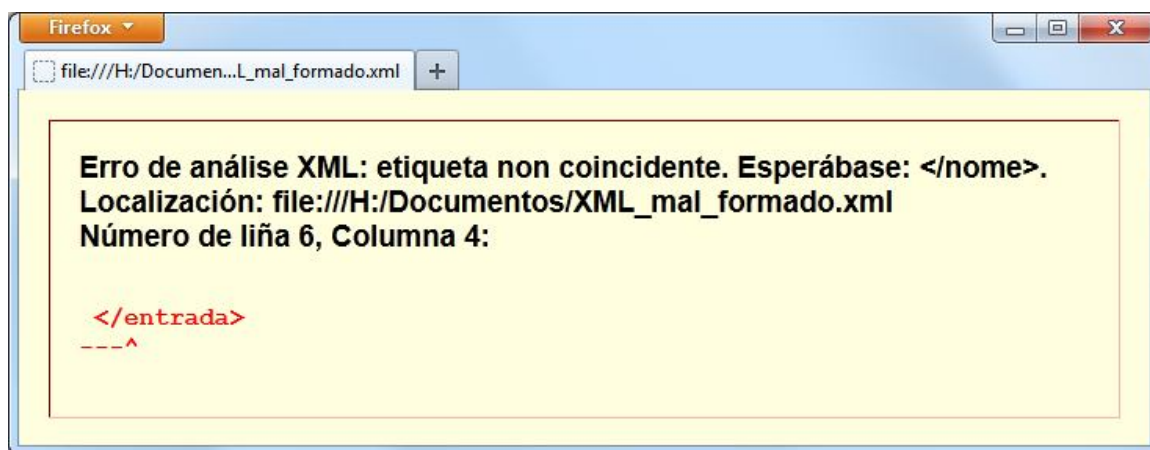
Na actualidade, os navegadores poden distinguir os documentos ben formados dos documentos mal formados. Se abrimos un documento XML nun navegador, se o documento está ben formado, veremos a información que contén en forma de árbore con nodos dinámicos. Se o documento ten erros, o navegador sinalará o primeiro elemento que conteña o erro.


**Imos ver como se visualiza un documento ben formado no navegador:**

```
<?xml version="1.1" encoding="ISO-8859-1"?>
<directorio>
  <entrada>
    <apelidos>Louzán Ferreiro</apelidos>
    <nome>Xaime</nome>
  </entrada>
</directorio>
```



Se modificamos o documento anterior eliminando a etiqueta de peche </nome> obteríamos un documento mal formado:



 Na tarefa 2 imos corrixir os erros que atopemos nalgúns documentos XML para acadar documentos ben formados.

## 3.4 Espazos en branco

XML define un conxunto de caracteres como "espazos en branco", que os procesadores permiten para facer máis lexible o código (por exemplo, desprazar as etiquetas para reflectir o aninamento). Estes caracteres son:

- Espazos (32).
- Tabuladores (9).
- Saltos de liña (10).
- Retornos de carro (13).

Normalmente as aplicacións que procesan os documentos XML eliminan do seu contido os espazos sobrantes. Isto é, o contido efectivo do elemento "apelidos" unha vez procesado sería o mesmo poñendo:

```
<apelidos>Louzán Ferreiro</apelidos>
```

Ou poñendo:

```
<apelidos>
  Louzán Ferreiro
</apelidos>
```

Sen embargo, nalgúns casos queremos indicarlle á aplicación que procese o documento que preserve os espazos tal e como figuran no documento (por exemplo, cando o documento contén unha poesía). Nestes casos, teremos que empregar un atributo especial de XML: "xml:space", que pode ter dous valores:

- "default". Indica á aplicación que procese o documento que trate os espazos do xeito que o fai habitualmente.
- "preserve". Indica á aplicación que debe preservar os espazos que figuran no contido do elemento tal e como aparecen reflectidos no documento.

Por exemplo:

```
<autor>Rosalía de Castro</autor>
<poesía xml:space="preserve">
Adiós, ríos; adios, fontes;
adios, regatos pequenos;
adios, vista dos meus ollos:
non sei cando nos veremos.
</poesía>
```

## 3.5 Identificación da linguaxe do contido

Existe outro atributo especial definido na especificación de XML que se emprega para identificar o idioma no que está escrito o contido dun elemento: "xml:lang". O seu valor é un código de dúas letras dos que recolle a norma ISO 639-1 (por exemplo, "gl" para o galego e "es" para o castelán).

Un documento XML que empregue este atributo podería ser:

```
<canción título="Automatic imperfection" xml:space="preserve"
xml:lang="en">

There are children playing on the manpost
'tending they are kids
They can climb up and fall down hard on their knees
...
</canción>

<canción título="Lo que sueñas vuela" xml:space="preserve" xml:lang="es">
Caminando sin pensar,
despacito, sin tiempo que ganar
...
</canción>
```

## 3.6 As entidades

As entidades son estruturas XML cun nome asociado. Para facer referencia a unha entidade emprégase a seguinte sintaxe:

&nomeDaEntidade;

Ao referenciar unha entidade polo seu nome insírese automaticamente o seu contido no canto da súa referencia. Xa vimos algunha anteriormente, e na seguinte unidade comentaremos cómo crear entidades, pero polo de agora veremos qué entidades están predefinidas na especificación de XML:

- "&amp;" para facer referencia ao carácter "&".
- "&lt;" para facer referencia ao carácter "<".
- "&gt;" para facer referencia ao carácter ">".
- "&quot;" para facer referencia á comiña dobre "\".
- "&apos;" para facer referencia á comiña simple "'".

## 3.7 A sección CDATA

Ás veces o uso das entidades pode ser moi incómodo, e facer a lectura do documento moi complicada. Por iso en ocasións pode ser moi útil utilizar unha sección "`<![CDATA[ ... ]]>`".

A sección CDATA ábrese cos caracteres "`<![CDATA[`" e péchase con "`]]>`". O contido que vai dentro dunha sección CDATA é ignorado polo analizador, co que podemos empregar caracteres como "<" e "&".

Por exemplo, se quixéramos engadir nun elemento XML contido en linguaxe HTML como o seguinte:

```
<H1>Antony & The Johnsons</H1>
```

Empregando entidades teríamos que facer:

```
<autor>&lt;/H1&gt;Antony &amp; The Johnsons&lt;/H1&gt;</autor>
```

Ao empregar unha sección CDATA o contido resulta máis claro:

```
<autor><![CDATA[<H1>Antony & The Johnsons</H1>]]></autor>
```

A única limitación lóxica ao seu contido é que dentro dunha sección CDATA non pode aparecer a cadea de peche "`]]>`". Neste caso deberase substituír por "`]]&gt;`".



Na tarefa 3 imos crear un novo documento XML.

## 4. Necesidade dos espazos de nomes

Un espazo de nomes é unha forma de agrupar elementos e atributos cunha orixe común, para diferencialos doutros elementos do mesmo nome.

Por exemplo, supoñamos que unha tenda de alimentación crea unha gramática XML para os produtos que vende, de xeito que un documento XML típico podería ser:

```
<?xml version="1.1" encoding="utf-8"?>
<produtos>
  <produto>
    <cod>LACT02330993</cod>
    <descricao>Leite enteira envase 1L</descricao>
  </produto>
  <produto>
    <cod>LACT00493112</cod>
    <descricao>Margarina vexetal tarrina 250g</descricao>
  </produto>
  ...
</produtos>
```

E máis adiante a mesma empresa crea unha nova gramática para os clientes habituais; por exemplo:

```
<?xml version="1.1" encoding="utf-8"?>
```

```

<clientes>
  <cliente>
    <cod>CL09384</cod>
    <nome>Uxío Fuentes Neira</nome>
    <endereço>Rúa Europa 24, 3ºA</endereço>
    <teléfono>555098433</teléfono>
  </cliente>
  <cliente>
    ...
  </cliente>
  ...
</clientes>

```

En ámbolos dous documentos existe un elemento de nome "cod". Mentres os datos de clientes non se mesturen cos datos dos produtos, non haberá problema. Pero se necesitáramos crear documentos para as vendas, nos que figuren os datos dos clientes e os datos dos produtos que mercan, teríamos que diferenciar dalgún xeito os elementos "cod" segundo o seu significado.

O xeito máis sinxelo de facelo sería cambiando o nome dun (ou ambos) destes elementos, poñendo por exemplo "cod\_producto" e "cod\_cliente". Pero en moitos casos esta solución non é válida; por exemplo, habería que modificar as aplicacións que están funcionando co nome orixinal dos elementos.

Os espazos de nomes son unha recomendación W3C (*World Wide Web Consortium*) para que os nomes comúns de elementos e etiquetas non colisionen, ben porque proveñen de distintos documentos XML a un posible documento destino, ben porque nos interesa separar dun xeito estruturado o contido dun documento. No exemplo anterior poderíamos empregar dous espazos de nomes: un para os produtos e outro para os clientes; deste xeito soamente temos que asegurarnos de que cada nome é único dentro do seu espazo de nomes. Cando xuntemos ambos elementos "cod", cada un deles estará asociado ao seu respectivo espazo de nomes e non colisionarán.

Debemos empregar espazos de nomes nos nosos documentos XML se existe a posibilidade de que sexan compartidos e mesturados con outros documentos XML.

## 4.1 Declaración dun espazo de nomes

Para declarar un espazo de nomes, emprégase a palabra reservada "xmlns" (*XML Namespace*). Por exemplo:

```

<?xml version="1.1" encoding="utf-8"?>
<produtos xmlns="http://www.atendadepaco.com/espazosdenomes/produtos/">
  <produto>
    <cod>LACT02330993</cod>
    <descrición>Leite enteira envase 1L</descrición>
  </produto>
  <produto>
    <cod>LACT00493112</cod>
    <descrición>Margarina vexetal tarrina 250g</descrición>
  </produto>
  ...
</produtos>

```

Cando declaramos un espazo de nomes no elemento raíz deste xeito, tódolos elementos do documento pasan a formar parte do espazo de nomes. É o que se chama un *espazo de nomes por defecto*. Os nomes dos elementos do documento pasan a estar compostos de dúas partes: o espazo de nomes + o nome local. Os nomes locais por si mesmos xa non identifican aos elementos do documento.

É importante destacar que as declaracións de espazos de nomes empregan a mesma sintaxe que os atributos, pero non se consideran atributos.

## URIs, URLs e URNs

Para identificar un espazo de nomes, e diferencialo doutros, asóciase cunha cadea de texto. Esta cadea de texto debe única, isto é, non poderá empregarse para identificar outros espazos de nomes.

- Unha solución é empregar unha URL (*Uniform Resource Locator*, Localizador Uniforme de Recurso) para a cadea de texto do espazo de nomes. *Unha URL especifica a localización dun recurso*, por exemplo, un arquivo nun servidor FTP ou unha páxina web:

`http://www.atendadepaco.com/espazosdenomes/produtos/`

Nestes casos o recurso referenciado pola URL soe conter información acerca do espazo de nomes.

- Unha URL é unha forma especial dunha URI (*Uniform Resource Identifier*, Identificador Uniforme de Recurso). *Unha URI é soamente un identificador; non ten que ser a dirección dun recurso existente*. Pode tomar a forma dunha URL ou calquera outra.

O xeito máis común de identificar un espazo de nomes é empregando unha URI coa forma dunha URL. Por exemplo, se o recurso identificado pola URL anterior non existise, a dirección seguiría a ser válida como identificador (URI) do espazo de nomes:

`http://www.atendadepaco.com/espazosdenomes/produtos/`

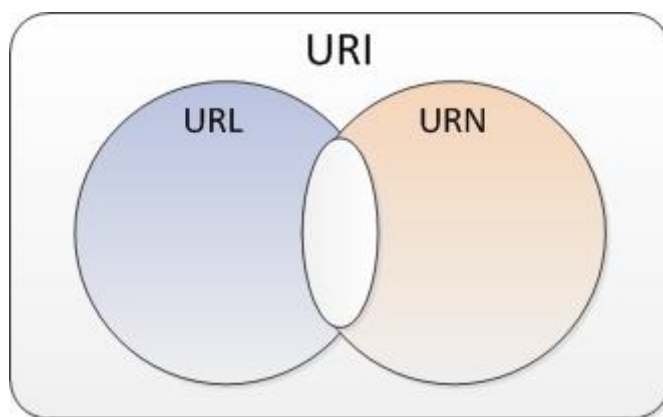
- Outro tipo de URI é unha URN (*Uniform Resource Name*, Nome Uniforme de Recurso). Unha URN é un nome que identifica algo. Cada URN está formada por un espazo de nomes e unha cadea que identifica un ítem dese espazo de nomes. A súa forma é:

`urn:[espazo de nomes]:[cadea de texto]`

Os identificadores dos espazos de nomes para as URNs poden ser rexistrados na IANA (*Internet Assigned Numbers Authority*, Autoridad de Asignación de Números de Internet). Por exemplo, no espazo de nomes dos ISBN dos libros (`isbn`), unha URN podería ser:

`urn:isbn:9780470114872`

Resumindo, a cadea que identifique un espazo de nomes debe ser unha URI. Dous tipos especiais de URI son as URL e as URN, aínda que estas últimas empréganse pouco.



## 4.2 Espazos de nomes con prefixo

Como xa vimos, cando no elemento raíz dun documento empregamos un espazo de nomes por defecto, este afecta ao resto dos elementos do documento. Isto é útil no caso dos documentos que conteñen un único espazo de nomes, como os documentos de produtos ou de clientes cos que traballamos antes.

Cando necesitamos empregar máis de un espazo de nomes nun documento XML, debemos elixir un prefixo para eles. O prefixo indícase a continuación da palabra "xmlns", separado polo carácter ":", e será o que teremos que empregar con aqueles elementos do documento para reflectir que pertencen ao espazo de nomes determinado.

Por exemplo, se creáramos un documento de vendas, poderíamos facelo do seguinte xeito:

```
<?xml version="1.1" encoding="utf-8"?>
<venda xmlns="http://www.atendadepaco.com/espazosdenomes/vendas/"
xmlns:cl="http://www.atendadepaco.com/espazosdenomes/clientes/"
xmlns:pr="http://www.atendadepaco.com/espazosdenomes/produtos/">
  <cl:cliente>
    <cl:cod>CL09384</cl:cod>
    <cl:nome>Uxío Fuentes Neira</cl:nome>
    <cl:endereco>Rúa Europa 24, 3ºA</cl:endereco>
    <cl:teléfono>555098433</cl:teléfono>
  </cl:cliente>
  <pr:produtos>
    <pr:produto>
      <pr:cod>LACT02330993</pr:cod>
      <pr:descrición>Leite enteira envase 1L</pr:descrición>
    </pr:produto>
    <pr:produto>
      <pr:cod>LACT00493112</pr:cod>
      <pr:descrición>Margarina vexetal tarrina 250g</pr:descrición>
    </pr:produto>
    ...
  </pr:produtos>
  ...
  <importe_total>16,34€</importe_total>
</venda>
```

Isto é:



- O **espazo de nomes por defecto** para o documento é o correspondente aos documentos de venda ("http://www.atendadepaco.com/espazosdenomes/vendas/").
- No elemento raíz se declaran tamén **outros dos espazos de nomes**, indicando un **prefixo para eles**, e se empregan indicando ese prefixo nos elementos do documento.
- Cando se emprega un prefixo nun elemento, debe facerse tanto na etiqueta de apertura como na etiqueta de peche.
- Deberemos indicar o prefixo en tódolos elementos pertencentes ao espazo de nomes. Por exemplo, o espazo de nomes "http://www.atendadepaco.com/espazosdenomes/produtos/" debemos aplicalo tanto ao elemento "produtos", como tamén aos subelementos "produto", "descripción" e "cod".
- Os nomes dos prefixos deben cumprir as mesmas normas que os nomes dos elementos e atributos, e ademais tampouco poden empregar o carácter ":".

## 4.3 Ámbito dun espazo de nomes

O ámbito dun espazo de nomes é o conxunto de elementos do documento no que podemos empregalo.

Cando declaramos un espazo de nomes no elemento raíz, o seu ámbito é o documento **enteiro**; poderemos empregar ese espazo de nomes en calquera elemento do documento, incluído o elemento raíz.

Pero aínda que é unha práctica común declarar tódolos espazos de nomes dun documento no seu elemento raíz, isto non é preciso e en moitos casos tampouco é doado. Dentro dun documento XML, pódense declarar espazos de nomes (por defecto ou con prefixo) en calquera dos seus elementos. O ámbito dun espazo de nomes abrangue o elemento no que se declara, e tódolos elementos que este contén.

É imprescindible declarar un espazo de nomes no mesmo elemento no que o imos a **empregar**, ou nun elemento **pai**. Por exemplo, poderíamos ter feito o documento de venda anterior do seguinte xeito:

```
<?xml version="1.1" encoding="utf-8"?>
<venda xmlns="http://www.atendadepaco.com/espazosdenomes/vendas/">
  <cl:cliente
    xmlns:cl="http://www.atendadepaco.com/espazosdenomes/clientes/">
    <cl:cod>CL09384</cl:cod>
    <cl:nome>Uxío Fuentes Neira</cl:nome>
    <cl:endereco>Rúa Europa 24, 3ºA</cl:endereco>
    <cl:teléfono>555098433</cl:teléfono>
  </cl:cliente>
  <pr:produtos
    xmlns:pr="http://www.atendadepaco.com/espazosdenomes/produtos/">
    <pr:produto>
      <pr:cod>LACT02330993</pr:cod>
      <pr:descripción>Leite enteira envase 1L</pr:descripción>
    </pr:produto>
    <pr:produto>
      <pr:cod>LACT00493112</pr:cod>
      <pr:descripción>Margarina vexetal tarrina 250g</pr:descripción>
    </pr:produto>
  ...
</venda>
```

```

</pr:produtos>
...
<importe_total>16,34€</importe_total>
</venda>

```

Podemos redefinir o espazo de nomes por defecto, ou un espazo de nomes asociado a un prefixo, dentro do seu ámbito. Para iso simplemente temos que definilo novamente. Por exemplo:

```

<?xml version="1.1" encoding="utf-8"?>
<venda xmlns="http://www.atendadepaco.com/espazosdenomes/vendas/">
  <cliente xmlns="http://www.atendadepaco.com/espazosdenomes/clientes/">
    <cod>CL09384</cod>
    <nome>Uxío Fuentes Neira</nome>
    <endereço>Rúa Europa 24, 3ºA</endereço>
    <teléfono>555098433</teléfono>
  </cliente>
  <produtos xmlns="http://www.atendadepaco.com/espazosdenomes/produtos/">
    <produto>
      <cod>LACT02330993</cod>
      <descricao>Leite enteira envase 1L</descricao>
    </produto>
    <produto>
      <cod>LACT00493112</cod>
      <descricao>Margarina vexetal tarrina 250g</descricao>
    </produto>
    ...
  </produtos>
  ...
  <importe_total>16,34€</importe_total>
</venda>

```

Tamén é posible eliminar a definición anterior dun espazo de nomes por defecto, ou tamén (a partires da versión 1.1 de XML) dun espazo de nomes con prefixo. Soamente temos que redefinilo empregando como identificador unha cadea baleira.

```

<?xml version="1.1" encoding="utf-8"?>
<elemento_raiz xmlns="http://www.exemplo.com/">
  <elemento1 xmlns="">
    ...
  </elemento1>
  ...
</elemento_raiz>

```

No exemplo anterior, o espazo de nomes por defecto declarado no elemento raíz non se aplica ao elemento "<elemento1>" nin a tódolos seus subelementos.

## 4.4 Atributos e espazos de nomes

Os espazos de nomes tal e como os vimos aplícanse soamente aos elementos. Non se aplican aos atributos, e os outros elementos dun documento XML, como os comentarios

ou as instrucións de procesamento, non teñen espazos de nomes. Por exemplo, no seguinte documento:

```
<?xml version="1.1" encoding="utf-8"?>
<elemento_raiz xmlns:ns="http://www.exemplo.com/">
  <ns:elemento1 cod="A334">
    ...
  </ns:elemento1>
  ...
</elemento_raiz>
```

O espazo de nomes "http://www.exemplo.com/" declárase no elemento raíz, e aplícase ao elemento "<elemento1>"; isto non inclúe ao atributo "cod". Nos documentos XML, os atributos pertencen a un elemento concreto, e isto evita que poidan colisionar con atributos pertencentes a outros espazos de nomes.

De calquera xeito, se quixéramos aplicar un espazo de nomes a un atributo, poderíámolo facer empregando a mesma sintaxe que cos elementos:

```
<?xml version="1.1" encoding="utf-8"?>
<elemento_raiz xmlns:ns="http://www.exemplo.com/">
  <ns:elemento1 ns:cod="A334">
    ...
  </ns:elemento1>
  ...
</elemento_raiz>
```



Na tarefa 4 empregaremos os espazos de nomes na creación de documentos XML.