

Recoñecemento das características de linguaxes de marcas. XML

1.	Introdución ás linguaxes de marcas	3
2.	Que é XML?	3
3.	Introdución á linguaxe XML	3
	Diferencias entre XML e HTML	5
	Vantaxes do emprego de XML	5
4.	Regras sintácticas de un documento XML	6
	I. Elemento raíz	6
	II. Prólogo	7
	III. Etiqueta de peche	7
	IV. Sensibles a maiúsculas e minúsculas	7
	V. Deben estar aniñados adecuadamente	7
	VI. Os valores dos atributos deben ir entre comiñas	8
	VII. Referencias as entidades	8
	VIII. Comentarios en XML	9
	IX. Conservación dos espazos en branco	9
	X. XML almacena as novas liñas como LF	9
5.	Contido dun documento XML	9
	5.1. Elementos	9
	Nomes dos elementos	10
	5.2. Atributos	11
	Atributos vs <i>subelementos</i>	11
	Valores dos atributos	12
	5.3. Texto dun documento XML	13
	5.4. Comentarios	14
	5.5. Instrucións de procesamento	14
6.	Documentos XML ben formados	14
	6.1. Especificacións de XML	16
	6.2. O prólogo	16
	6.3. Visualización de documentos XML en navegadores web	17
	6.4. Espazos en branco	19

6.5.	Identificación do idioma do contido	20
6.6.	As entidades.....	20
6.7.	A sección CDATA.....	21
7.	Necesidade dos espazos de nomes	22
8.	Declaración dun espazo de nomes.....	23
8.1.	URI, URL e URN.....	24
8.2.	Espazos de nomes con prefixo	25
8.3.	Ámbito dun espazo de nomes.....	27
8.4.	Atributos e espazos de nomes.....	29

A linguaxe XML

1. Introducción ás linguaxes de marcas

Nesta unidade traballaranse os seguintes conceptos e manexo de destrezas:

- Identificar os ámbitos de aplicación da linguaxe XML e as súas vantaxes.
- Identificar a estrutura dun documento XML e as súas regras sintácticas.
- Crear documentos XML ben formados.
- Identificar as vantaxes que aporta o emprego dos espazos de nomes.
- Escoller e declarar espazos de nomes.
- Crear documentos XML que empreguen espazos de nomes.

2. Que é XML?

- XML significa **eXtensible Markup Language**.
- XML foi deseñado para **almacenar e transferir datos**, tanto para persoas como *computadoras* ou Sistemas de lectura a máquina.

3

```
<?xml version="1.0" encoding="UTF-8"?>
<correo>
  <para>fernando.simon@covid19.es</para>
  <de>ayuso@comunidaddemadrid.es</de>
  <asunto>Confinamiento</asunto>
  <mensaxe>Desconfina!</mensaxe>
</correo>
```

XML ten **un papel moi importante para moitos diferentes sistemas de tecnoloxías da información**.

XML se **emprega moito para distribuír datos a través de Internet**.

Por todo iso, é moi importante o seu coñecemento. 😊

3. Introducción á linguaxe XML

A característica principal dunha linguaxe de marcas é que **engade ó 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>Páxina de LMSXI</title>
  </head>
  <body>
    <!-- corpo de HTML -->
    <h1>XML</h1>
    <p>XML é umha linguaxe sinxela.</p>
  </body>
</html>
```

Na linguaxe *HTML* existe un número de etiquetas predefinidas (*html*, *head*, *body*, *table*, *br*, ...). 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 un correo electrónico podería ser:

4

```
<?xml version="1.0" encoding="UTF-8"?>
<correo>
  <para>fernando.simon@covid19.es</para>
  <de>ayuso@comunidaddemadrid.es</de>
  <asunto>Confinamiento</asunto>
  <mensaxe>Desconfina!</mensaxe>
</correo>
```

O documento anterior, é auto descritivo, e contén información do remitente, destinatario, asunto, e mensaxe. Non ten información sobre a súa representación.

E un documento XML para os módulos dun ciclo:

```
<módulos>
  <módulo>Linguaxes de Marcas
    <sesións>5</sesións>
  </módulo>
  <módulo>Programación
    <sesións>9</sesións>
  </módulo>
</módulos>
```

Isto é, XML (*eXtensible Markup Language*, Linguaxe de Marcado Extensible) non é unha linguaxe de marcas propiamente dita. **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.

Diferencias entre XML e HTML

- XML está pensado para **transportar datos**, facendo énfase de como son os **datos**.
- **HTML** foi deseñada para **amosar datos**, con enfoque de **como se amosan os datos**.
- As etiquetas **XML non son etiquetas predefinidas**, como HTML.

5

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 de modo informático 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, pode ser o Notepad++ (mellor co plugin de XML), XML Copy Editor ou algún IDE máis avanzado, como Netbeans, Eclipse, Visual Studio Code, etc.

4. Regras sintácticas dun documento XML

As regras son poucas e sinxelas. Se cumpre estas regras se di que o documento está “**ben formado**”:

I. Elemento raíz

6

Todos os documentos XML **deben ter un elemento raíz** que é pai dos outros elementos. Por exemplo:

```
<raiz>
  <fillo>
    <neto>.....</neto>
  </fillo>
</raiz>
```

No seguinte exemplo o documento raíz é **<correo>**:

```
<?xml version="1.0" encoding="UTF-8"?>
<correo>
  <para>fernando.simon@covid19.es</para>
  <de>ayuso@comunidaddemadrid.es</de>
  <asunto>Confinamiento</asunto>
  <mensaxe>Desconfina!</mensaxe>
</correo>
```

II. Prólogo

A seguinte liña se chama **prólogo**:

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

- O prólogo é **OPCIONAL**.
- Se existe **debe ser o primeiro no documento**.
- Os documentos XML poden conter **caracteres internacionais**, como galego *ñóáéí*, francés *êêé* ou noruegués *øæå*, por exemplo. Para evitar erros, se **pode indicar o tipo de codificación empregado, ou poñelo como UTF-8**. Se fai co atributo *encoding*.
- A **codificación por defecto** dos documentos XML é **UTF-8**, como sucede en HTML5, CSS, JavaScript, PHP ou SQL.

III. Etiqueta de peche

- Os documentos XML debe ter unha **etiqueta de peche, obrigatoria**. É ilegal omitir unha etiqueta de peche.
- Ollo: **o prólogo non ten etiqueta de peche, pois non forma parte do documento**.
- Exemplos:

```
<mensaxe>...</mensaxe>  
<salto />
```

7

IV. Sensibles a maiúsculas e minúsculas

- As **etiquetas XML son sensibles a maiúsculas e minúsculas**. Por exemplo, a etiqueta `<Virus>` é diferente da etiqueta `<virus>`.
- As etiquetas de inicio e peche deben estar escritos iguais. Por exemplo:

```
<nota>A vida é breve</nota>
```

Aclaración: soen chamarse etiquetas de “inicio e fin” ou de “apertura” e “peche”.

V. Deben estar aniñados adecuadamente

- A diferenza de outras linguaxe como HTML, en XML **todos os elementos deben estar aniñados un dentro de outro**:

```
<nota><titulo>Viva a vida</titulo>. A veces</nota>
```

Deben **pecharse en orde inverso á apertura**. No exemplo anterior, o elemento “titulo” foi aberto dentro de “nota”, polo que debe pecharse antes.

VI. Os valores dos atributos deben ir entre comiñas

- Os elementos XML poden ter atributos con pares nome/valor, como HTML. **Os valores destes atributos deben ir sempre entre comiñas:**

```
<estudiante dni="52454443J">
  <nota>10</nota>
  <curso>1 DAW</curso>
</estudiante>
```

VII. Referencias as entidades

- Algúns **caracteres (letras) teñen un significado especial** en XML.
- Por exemplo, o carácter “<” dentro dun elemento daría un erro, posto que o *parser* o interpreta como inicio de un elemento:

```
<expresión>x + y < 3</expresión>
```

Para evitar este erro, debe substituírse “<” por unha referencia de entidade:

```
<expresión>x + y &lt; 3</expresión>
```

- Por defecto, en XML existen 5 referencias de entidades predefinidas:

&lt;	<	Menor que
&gt;	>	Maior que
&amp;	&	<i>ampersand</i>
&apos;	'	Apóstrofe
&quot;	"	Comiñas

Estritamente, só < e & son ilegais en XML, pero é boa recomendación substituír tamén > por >

VIII. Comentarios en XML

A sintaxe para escribir comentarios en XML é **similar a HTML**, con `<!-- ... -->` (*dobres guións despois e antes de < e >, respectivamente*):

```
<!-- Isto é un comentario -->
```

Non se permiten dobres guións dentro dun comentario:

```
<!--Este comentario -- NON é VÁLIDO -->
```

IX. Conservación dos espazos en branco

A diferenza de HML, **XML non trunca múltiples espazos en branco nun só espazo** (HTML trunca múltiples espazos, saltos de liña, ... como un único espazo):

XML:	Ola	mundo
HTML:	Ola mundo	

X. XML almacena as novas liñas como LF

En **Windows**, as aplicacións gardan as novas liñas como retorno de carro e salto de liña (**CR+LF**) [como as antigas máquinas de escribir].

XML garda os salto de liña como LF (line feed ou salto de liña, sen retorno de carro).

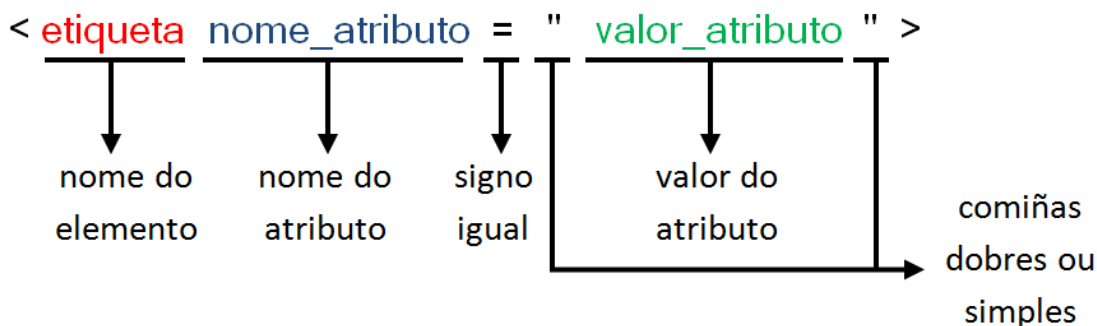
Así, por exemplo: en **Linux/Unix as novas liñas son LF** (como XML) e en **antigos SO Mac eran CR** (retorno de carro).

Cando un documento XML cumpre as regras anteriores se di que é un documento XML “Ben Formado”

5. Contido dun documento XML

5.1. 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. Poden ser varios, sen repeticións. Os valores dos atributos poden ir entre comiñas dobres ou sinxelas
- **texto**, entre a etiqueta de apertura e a etiqueta de peche.
- **outros elementos**. Poden conter outros elementos, que se deben abrir e pechar dentro do elemento.

Por exemplo, o elemento "libro" contén os atributos "editorial" e "isbn", un texto, e os elementos fillo "autora":

```
<libro editorial="La cúpula" isbn="9788415724728">Aquel verano
    <autora>Jillian Tamaki</autora>
    <autora>Mariko Tamaki</autora>
</libro>
```

Poden existir elementos baleiros, que poden representarse por dúas ou por unha única etiqueta:

```
<favorito></favorito>
<favorito />
```

Os elementos **sempre se deben pechar**.

Nomes dos elementos

O nome dun elemento figura na **etiqueta de apertura e de peche e 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**.

O primeiro carácter non pode ser un número nin algún dos caracteres " . " e "-".

Exemplos de nomes de elementos **non** serían válidos:

```
<19Covid></19Covid>
<+elemento></+elemento>
<.aparte></.aparte>
<-elemento></-elemento>
```

Aínda que, segundo se reflexa na especificación da linguaxe, tamén se pode empregar o carácter ":" nos nomes dos elementos, **débese evitar** pois pode confundirse cando empregamos **espazos de nomes**, que veremos máis adiante.

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". Non se poden empregar.

5.2. Atributos

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

11

```
<libro editorial="La cúpula" isbn="9788415724728">...</libro>
```

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 é relevante a orde na que estes figuren dentro do elemento.

IMPORTANTE: un elemento non pode ter atributos co nome repetido.

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

Atributos vs subelementos

Ás veces temos que **escoller entre empregar un subelemento ou un atributo para almacenar certa información**. Por exemplo, o ISBN dun libro se podería poñer como elemento:

```
<libro editorial="La cúpula"> Aquel verano
```

```
<isbn>Jillian Tamaki</isbn>
</libro>
```

Ou como atributo:

```
<libro editorial="La cúpula" isbn="9788415724728">Aquel verano
</libro>
```

Os atributos están pensados para gardar información (descricións) sobre os datos e só poden almacenar un único valor, mentres que os elementos poden almacenar varios datos e incluso estenderse mediante estruturas aniñadas de varios elementos.

Como no exemplo anterior o ISBN é único e informa sobre o libro (clave) mellor é empregalo como atributo.

Valores dos atributos

O valor do atributo debe ir **pechado 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 de comiñas 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, sempre deben levar "=" a un valor, aínda que as comiñas non conteñan datos. Non é válido:

```
<casiña seleccionada></casiña>
```

O correcto sería:

```
<casiña seleccionada="sí"></casiña>
```

Ou:

```
<casiña seleccionada=""></casiña>
```

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

A partir do seguinte código xml seguinte:

```

<colección musical>
  <artista nome=Queen>
    <album nome=A Night at the Opera>
      <Bohemian Rhapsody>
        <'39>
      </album>
    </artista>
  <artista nome=R.E.M.>
    <album nome='Life's Rich Pageant'>
      <Superman>
        <These days>
      </album>
    </artista>
  
```

Trátase de crear, a partires do documento anterior, un novo documento XML que conteña a mesma información pero que NON teña erros nos nomes e valores dos elementos e atributos.

5.3. Texto dun documento XML

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.

Como vimos nas regras, o texto dun documento XML **pode conter calquera tipo de carácter salvo "<" e "&"**. Por exemplo, non sería válido:

```
<autora>Mariko & Jillian Tamaki</autora>
```

A mesma **regra aplícase ós valores dos atributos**, que ademais dos caracteres "<" e "&" **tampouco poden conter o mesmo tipo de comiñas** que se emprega para delimitar o valor do atributo. Non é válido:

```
<actor nome='Peter O'Toole'></actor>
```

Para empregar estes caracteres especiais dentro do texto ou dentro do valor dun atributo hai 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 substituírse, respectivamente, polas referencias "&" e "<" (en *hexadecimal* serían "&" e "<")
- empregar **entidades**. Como veremos máis adiante, e adiantamos no apartado das regras, empregando entidades poderíamos substituír os caracteres "&" e "<" respectivamente polas cadeas "&" e "<".

Así, os exemplos anteriores quedaerían:

```
<autora>Mariko &amp; Jillian Tamaki</autora>
```

```
<actor nome='Peter O&#38;Toole'></actor>
```

5.4. 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 "`-->`".

```
<!-- Aquí comeza o importante -->
```

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

5.5. Instrucións de procesamento

As instrucións de procesamento **empregáanse 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"?>
```

Veremos polo miúdo instrucións de procesamento na unidade que trata a **linguaxe XSLT**.

6. Documentos XML ben formados

Un documento considérase "*ben formado*" se respecta as normas ou regras XML, e tamén as seguintes descritas na especificación oficial de XML que xa se mencionaron con anterioridade:

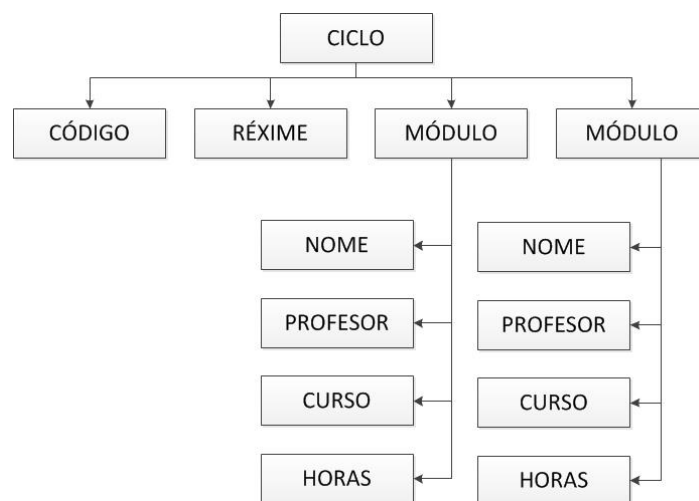
- Tódolos elementos do documento **aníñanse en forma de árbore**, existindo un **único elemento raíz**, que contén ao resto de elementos. Por exemplo:

```
<ciclo>
  <código>DAW</código>
```

```

<r  xime>Ordinario</r  xime>
<m  dulo>
  <nome>Linguaxes de marcas e sistemas de informaci  n</nome>
  <profesor>Fernando Sim  n</profesor>
  <curso>1.  </curso>
  <horas>133</horas>
</m  dulo>
<m  dulo>
  <nome>Programaci  n</nome>
  <profesor>Isabel Ayuso</profesor>
  <curso>1.  </curso>
  <horas>250</horas>
</m  dulo>
</ciclo>

```



- **T  dolos elementos deben pecharse**, e isto debe facerse sempre de forma estruturada,    dicir, en orde inversa a s  a apertura: o   ltimo elemento en abrirse ser   o primeiro en pecharse.

```

<libro editorial="La c  pula" isbn="9788415724728">Aquel verano
  <autora>Jillian Tamaki
    <idade>35</idade>
  </autora>
  <autora>Mariko Tamaki</autora>
</libro>

```

- A etiqueta de peche dun elemento debe coincidir exactamente co nome que figura na etiqueta de apertura, incl  ndo min  sculas e mai  sculas.
- Na especificaci  n **1.1 da linguaxe**, os documentos XML deben comezar cun pr  logo.

6.1. Especificacións de XML

Existen **dúas especificacións da linguaxe XML: a 1.0 e a 1.1**, con poucas diferenzas (algún editor ou plugin, como no Notepad++, non da soporte para a versión 1.1, aínda así, tamén se emprega).

Novidades para a 1.1:

- Soporte **Unicode completo**. A especificación 1.0 límtase aos caracteres recollidos en Unicode 2.0, mentres que a especificación 1.1 abrangue incluso caracteres de versións posteriores.
- A capacidade de recoñecer outros caracteres de fin de liña. En XML 1.0 se pode empregar "
" para o salto de liña e "" para o retorno de carro. A versión 1.1 engade **soporte para os caracteres de fin de liña definidos por Unicode** (" " e por EBCDIC ("…").
- É **obligatorio engadir un prólogo** ós documentos para que estean ben formados.

6.2. 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:

16

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

Opcionalmente pode incluír os atributos:

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

Tódolos procesadores XML **deben ser capaces de ler documentos codificados en UTF-8 e UTF-16** e deducir cal de ambos é o empregado nun documento. Se é posible detección de codificación, non é preciso pero si aconsellable indicar o atributo `encoding`.

Nota: 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"?>
```

Nota: algúns navegadores Web antigos non procesan correctamente os documentos XML que conteñan un prólogo.

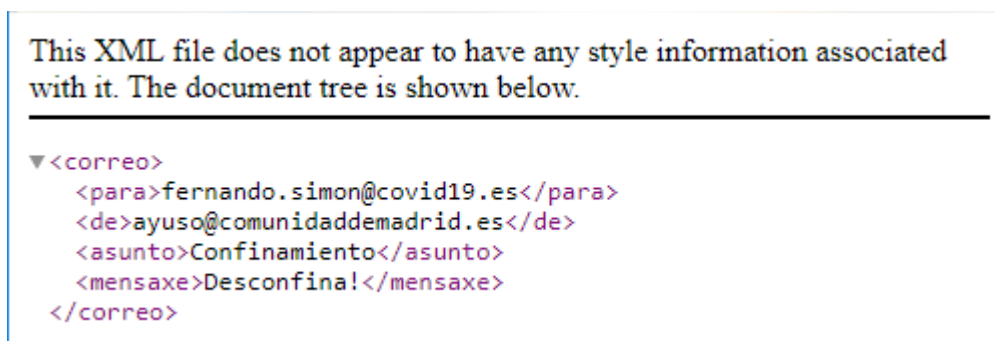
6.3. Visualización de documentos XML en navegadores web

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.

O documento:

```
<?xml version="1.1" encoding="UTF-8"?>
<correo>
  <para>fernando.simon@covid19.es</para>
  <de>ayuso@comunidaddemadrid.es</de>
  <asunto>Confinamiento</asunto>
  <mensaxe>Desconfina!</mensaxe>
</correo>
```

Pode verse:



Se elimina o carácter / da etiqueta de peche de mensaxe aparece un erro:

This page contains the following errors:

warning on line 1 at column 20: Unsupported version '1.1'
 error on line 7 at column 10: Opening and ending tag mismatch: mensaxe line 0 and correo

Below is a rendering of the page up to the first error.

fernando.simon@covid19.es ayuso@comunidaddemadrid.es Confinamiento Desconfina!

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

Corrixe os erros que atopes nos seguintes documentos XML, para obter documentos XML ben formados segundo a especificación 1.1 da linguaxe:

Documento 1

```
<!-- colección_musical.xml ->
<xml versión="1.1" encoding="utf-8">
<artista nome="Queen">
  <album nome="A Night at the Opera">
    <canCIÓN>Bohemian Rhapsody</Canción>
    <canCIÓN>'39</Canción>
  </album>
</artista>
<artista nome="R.E.M.">
  <album nome="Life's Rich Pageant">
    <canCIÓN>Superman</Canción>
    <canCIÓN>These days</Canción>
  </album>
</artista>
<artista nome="Iggy Pop">
```

```

    <album nome="Brick by Brick">
      <canción>Candy</Canción>
      <canción>My Baby Wants to Rock & Roll</Canción>
    </artista>
  </album>

```

Documento 2

```

<?xml version="1.1" encoding="utf-8">
<programas>
  <2008-2013>
    Programas con data > 2008 e < 2013
    <programa nome="Notepad++" licenza=GPL licenza=LGPL Windows />
    </programa>
    <programa nome="XML Copy Editor" licenza=GPL Windows Linux />
    </programa>
  </2008-2013>
</programas>

```

6.4. Espazos en branco

XML define un conxunto de caracteres como "espazos en branco", que os procesadores permiten para **facen máis lexible o código** (por exemplo, tabular as etiquetas para reflectir pais/fillos). Estes caracteres son:

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

19

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>Simón Soria</apelidos>
```

Ou poñendo:

```

<apelidos>
  Simón Soria
</apelidos>

```

Se queremos indicarlle á aplicación que procese o documento que **preserve os espazos tal e como figuran no documento** 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. Por defecto.
- **"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>
```

6.5. Identificación do idioma 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):

https://www.loc.gov/standards/iso639-2/php/English_list.php
https://es.wikipedia.org/wiki/ISO_639-1

20

Exemplo:

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

6.6. As entidades

As entidades son estruturas XML cun **nome asociado** (como unha variable -> valor dunha linguaxe de programación). Para facer referencia a unha entidade emprégase a seguinte sintaxe:

&nomeDaEntidade;

Ó referenciar unha entidade polo seu nome **insírese automaticamente o seu contido** no canto da súa referencia. Na seguinte unidade se verá como declarar entidades.

Existen entidades predefinidas en XML:

- `"&";` para facer referencia ao carácter "&".
- `"<";` para facer referencia ao carácter "<".
- `">";` para facer referencia ao carácter ">".
- `""";` para facer referencia á comiñas dobre "\".
- `"'";` para facer referencia á comiñas simple "'".

6.7. 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 "&"**.

21

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 & The Johnsons&lt;/h1&gt;</autor>
```

Ó empregar unha sección CDATA o contido resulta máis claro:

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

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

Na tarefa 3 imos crear un novo documento XML.

Trátase de crear un documento XML sobre os libros dunha biblioteca cos seguintes datos:

- Para cada libro deberá aparecer o ISBN, o título, o ano de publicación e a editorial.
- Tamén deberá figurar o nome do autor ou autores (poderán ser máis de un).
- Cada libro pode ter unha ou varias descrições (ou ningunha). En caso de telas, deberase indicar o idioma no que está escrita.

O documento deberá conter información cando menos de 2 libros e se deberá comprobar que está ben formado seguindo a especificación XML 1.1.

7. Necesidade dos espazos de nomes

Un espazo de nomes é unha forma **de agrupar elementos e atributos cunha orixe común**, para **diferenciarlos 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 250g</descricao>
  </produto>
  ...
</produtos>
```

22

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 co mesmo 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 (o 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 entren en colisión** (*cando existen dous nomes iguais no mesmo documento pero con diferentes significados*), ben porque **proveñen de distintos documentos** XML a un posible documento destino ou ben porque interesa **separar dun xeito estruturado o contido dun documento**.

23

No exemplo anterior se podería empregar dous espazos de nomes: un para os produtos e outro para os clientes; deste xeito soamente se ten que asegurar 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 entrarán en colisión.

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

8. 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.pepinho.com/ns/produtos/">
  <produto>
    <cod>LACT02330993</cod>
    <descricao>Leite enteira envase 1L</descricao>
  </produto>
</produtos>

```

```

    </produto>
    <produto>
      <cod>LACT00493112</cod>
      <descricao>Margarina vexetal 250g</descricao>
    </produto>
    ...
  </produtos>

```

Cando **declaramos un espazo de nomes no elemento raíz**, 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**.

8.1. URI, URL e URN

24

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 boa 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, unha páxina Web:

```
http://www.pepinho.com/espazosdenomes/receitas/
```

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.pepinho.com/espazosdenomes/receitas/>

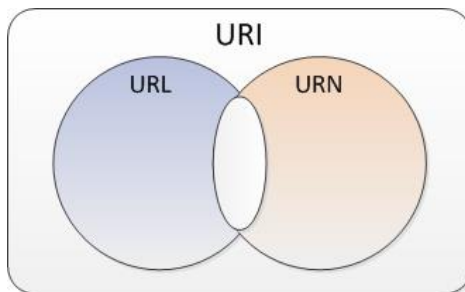
- Outro tipo de URI é unha **URN** (*Uniform Resource Name*, Nome Uniforme de Recurso). Unha URN **é un nome que identifica algo**. Cada URN está formada por un **espazo de nomes e unha cadea que identifica un ítem dese espazo de nomes**. A súa forma é:

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*

En resumo, **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. Para nós, o máis cómodo é unha URL (que non ten porque existir).



8.2. Espazos de nomes con prefixo

Como xa vimos, **cando no elemento raíz dun documento empregamos un espazo de nomes por defecto**, este afecta ó 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.pepinho.com/espazosdenomes/vendas/"
xmlns:cl="http://www.pepinho.com/espazosdenomes/clientes/"
xmlns:pr="http://www.pepinho.com/espazosdenomes/produtos/">
  <cl:cliente>
    <cl:cod>CL09384</cl:cod>
    <cl:nome>Maruja Mones</cl:nome>
    <cl:endereço>Rúa Morte 13, 40ºA</cl:endereço>
    <cl:teléfono>555555132</cl:teléfono>
  </cl:cliente>
  <pr:produtos>
    <pr:produto>
      <pr:cod>CHURR2330993</pr:cod>
      <pr:descricao>Bolsa Churros 1KG</pr:descricao>
    </pr:produto>
    <pr:produto>
      <pr:cod>PAN00493112</pr:cod>
      <pr:descricao>Pan de centeo 300gr</pr:descricao>
    </pr:produto>
    ...
  </pr:produtos>
  ...
  <importe_total>13,13€</importe_total>
</venda>

```

Isto é:

- O **espazo de nomes por defecto** para o documento é o correspondente aos documentos de venda ("http://www.pepinho.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.pepinho.com/espazosdenomes/produtos/" **debemos aplicalo tanto ao elemento "produtos", como tamén aos subelementos "produto", "descricao" 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 ":".

8.3. 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 inteiro**; 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 **abrange 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.pepinho.com/espazosdenomes/vendas/">
  <cl:cliente
    xmlns:cl="http://www.pepinho.com/espazosdenomes/clientes/">
    <cl:cod>CL09384</cl:cod>
    <cl:nome>Maruja Mones</cl:nome>
    <cl:endereço>Rúa Morte 13, 40°A</cl:endereço>
    <cl:teléfono>555555132</cl:teléfono>
  </cl:cliente>
  <pr:produtos
    xmlns:pr="http://www.pepinho.com/espazosdenomes/produtos/">
    <pr:produto>
      <pr:cod>CHURR2330993</pr:cod>
      <pr:descricao>Bolsa Churros 1KG</pr:descricao>
    </pr:produto>
    <pr:produto>
      <pr:cod>PAN00493112</pr:cod>
      <pr:descricao>Pan de centeo 300gr</pr:descricao>
    </pr:produto>
    ...
  </pr:produtos>
  ...
</importe_total>13,13€</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.pepinho.com/espazosdenomes/vendas/">
  <cliente xmlns="http://www.pepinho.com/espazosdenomes/clientes/">
    <cl:cod>CL09384</cl:cod>
    <cl:nome>Maruja Mones</cl:nome>
    <cl:enderezo>Rúa Morte 13, 40ºA</cl:enderezo>
    <cl:teléfono>555555132</cl:teléfono>
  </cl:cliente>
  <produtos
xmlns="http://www.pepinho.com/espazosdenomes/produtos/">
    <pr:produto>
      <pr:cod>CHURR2330993</pr:cod>
      <pr:descrición>Bolsa Churros 1KG</pr:descrición>
    </pr:produto>
    <pr:produto>
      <pr:cod>PAN00493112</pr:cod>
      <pr:descrición>Pan de centeo 300gr</pr:descrición>
    </pr:produto>
    ...
  </pr:produtos>
  ...
  <importe_total>13,13€</importe_total>
</venda>
```

28

Tamén é posible **eliminar a definición anterior dun espazo de nomes por defecto**, ou tamén **(a partir 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.

8.4. Atributos e espazos de nomes

Os espazos de nomes tal e como os vimos **se aplican só ós elementos**. **Non se aplican ós 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 entrar en colisión con atributos pertencentes a outros espazos de nomes.

Se quixeramos 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.

Nun centro de ensino empregan documentos XML separados con información sobre os alumnos e os ciclos que se imparten.

- documento dos alumnos ten a seguinte aparencia:

```

<?xml version="1.1" encoding="utf-8"?>
<!-- alumnos.xml -->
<alumnos>
  <alumno>
    <nome>Xoan</nome>
    <apelidos>Cancedo da Foz</apelidos>
    <ciclo>ASIR ordinario</ciclo>
    <curso>1º</curso>
    <datanacemento>06/11/1991</datanacemento>
    <nota>Xa fixo outro ciclo</nota>
  </alumno>
  <alumno>
    ...
  </alumno>
  ...
</alumnos>

```

- o documento coa información dos ciclos é o seguinte:

```

<?xml version="1.1" encoding="utf-8"?>
<!-- ciclos.xml -->
<ciclos>
  <ciclo id="ASIR" grao="CS">
    <módulo id="MP0370">
      <nome>Planificación e administración de redes</nome>
      <horas>213</horas>
    </módulo>
    <módulo id="MP0371">
      <nome>Fundamentos hardware</nome>
      <horas>107</horas>
    </módulo>
    ...
  </ciclo>
  <ciclo id="DAW" grao="CS">
    ...
  </ciclo>
  ...
</ciclos>

```

No centro queren xerar un novo tipo de documento XML, "centro.xml", empregando a gramática dos xa existentes e engadindo os elementos novos que sexan necesarios, tendo en conta que:

- O novo documento deberá conter para cada ciclo a mesma información do documento "ciclos.xml".
- Ademais dentro de cada elemento "ciclo" figurarán os alumnos matriculados nese ciclo.
- Cada un deses alumnos terá como subelementos "nome", "apelidos", "datanacemento" e "nota", tal e como figura no documento "alumnos.xml".
- Cada alumno terá tamén un elemento "módulo" por cada módulo no que estea matriculado.
- Estes elementos "módulo" terán soamente dous atributos: un atributo "id" co código do módulo, e outro atributo "nota" coa nota obtida polo alumno no módulo.

Empregar espazos de nomes para xerar un documento "centro.xml" coas indicacións anteriores, que conteña cando menos:

- Un ciclo con dous módulos.

- Nese ciclo haberá dous alumnos.
- Cada alumno estará matriculado nos dous módulos do ciclo.

