

## LMSGI01. RECOÑECIMENTO DAS CARACTERÍSTICAS DE LINGUAXES DE MARCAS



### Caso práctico

María e Félix son os fundadores e propietarios dunha asesoría legal e empresarial, que ten a súa sede en Cantabria, con oficinas nos municipios máis importantes da rexión.

María, licenciada en Dereito, exercía como avogada especializada en dereito laboral e representaba a algunha empresa, ademais de particulares no seu propio despacho situado en Torrelavega. Tiña unha rede informática cliente-servidor sobre un sistema operativo Windows e traballaba cunha base de datos de documentos xurídicos.

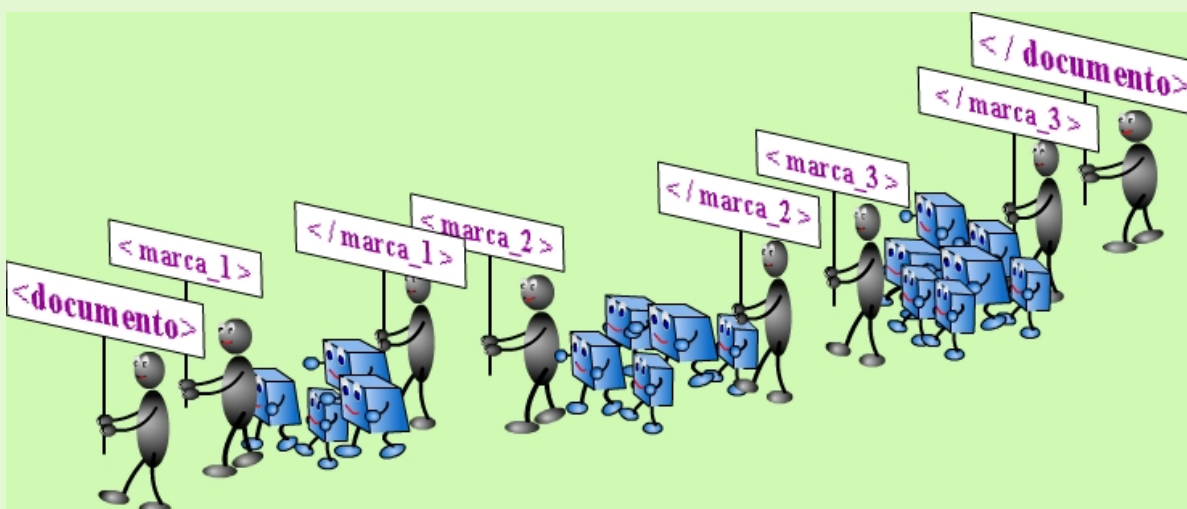
Félix, diplomado en Ciencias Empresariais, creara unha asesoría empresarial, situada en Santander, que basicamente, se encargaba da contabilidade de varias pemes. Tamén tiña unha rede cliente-servidor, pero esta baixo un sistema Linux con software libre de contabilidade.

Ambos eran amigos e un día no que estiveran falando dos seus respectivos traballos, decidiron que os seus ingresos podían aumentar substancialmente se, ademais de manter as súas respectivas carteiras de clientes, se unían e formaban unha sociedade que lles ofrecese ás empresas asesoría legal e empresarial de forma conxunta.

Desde o principio, a idea de asociarse foi un éxito. Ao cabo de dous anos o volume do negocio estendeuse e fíxose imprescindible o intercambio de comunicación entre ambos.

Dado que traballaban con sistemas informáticos diferentes presentábase o problema de como podían compartir información sobre os clientes comúns mantendo a infraestrutura informática coa que traballa cada un.

Consultáronlle o problema con Xoán, un técnico en administración de sistemas informáticos en rede, e este díxolles que non había ningún problema de interconexión se os ficheiros que manexaban se axustaban a un formato estándar coñecido como XML. Segundo o que Xoán lles dixo, xerar documentos co dito estándar apenas require coñecementos previos de informática, polo tanto era unha solución que parecía perfecta para o seu problema.



Uso educativo non comercial. Elaboración propia.

# 1. Linguaxes de marcas

Unha "linguaxe de marcas" **é un modo de codificar un documento** onde, xunto co texto, se incorporan etiquetas, marcas ou anotacións **con información adicional** relativa á estrutura do texto ou o seu formato de presentación. Permiten facer explícita a estrutura dun documento, o seu contido semántico ou calquera outra información lingüística ou extralingüística que se queira facer patente.

Toda linguaxe de marcas está definida nun documento denominado DTD (Document Type Definition). Nel establécense as marcas, os elementos utilizados pola dita linguaxe e as súas correspondentes etiquetas e atributos, a súa sintaxe e normas de uso.



## Exemplo

Aspecto dun documento realizado cunha linguaxe de marcas:

```
<carta>
  <data>22/11/2006</data>
  <presentacion>Estimado cliente:</presentacion>
  <contido>bla bla bla bla ...</contido>
  <firma>Don Xosé Gutiérrez González</firma>
</carta>
```

Aínda que na práctica, nun mesmo documento poden combinarse varios tipos diferentes de linguaxes de marca, estas pódense clasificar como segue:

- De presentación: define o formato do texto.
- De procedementos: orientado tamén á presentación pero, neste caso, o programa que representa o documento debe interpretar o código na mesma orde na que aparece.
- Descritivo ou semántico: describen as diferentes partes nas que se estrutura o documento pero sen especificar como se deben representar.

Algúns exemplos de linguaxes de marcado agrupadas polo seu ámbito de utilización son:

- Documentación electrónica
  - **RTF** (Rich Text Format): formato de texto enriquecido, foi desenvolvido por Microsoft en 1987. Permite o intercambio de documentos de texto ente 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 esta linguaxe.
  - **DocBook**: permite xerar documentos separando a estrutura lóxica do documento do seu formato. Deste xeito, os ditos documentos, poden publicarse 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 linguaxes especializadas
  - **MathML** (Mathematical Markup Language): o seu obxectivo é expresar o formalismo matemático de tal modo que poida ser entendido por distintos sistemas e aplicacións.
  - **VoiceXML** (Voice Extended Markup Language) ten como obxectivo o intercambio de información entre un usuario e unha aplicación con capacidade de recoñecemento de fala.
  - **MusicXML**: permite o intercambio de partituras entre distintos editores de partituras.



## Autoavaliación

As linguaxes de marcas utilízanse para:

- ☐ a) Dar formato aos documentos de texto.
- ☐ b) Definir a estrutura dos datos dun documento.
- ☐ c) Permitir o intercambio de ficheiros entre diferentes aplicacións e plataformas.
- ☐ d) Todas as anteriores.

## 2. Evolución das linguaxes de marcas

Nos anos 70 continuamente xorden unhas linguaxes informáticas, distintas das linguaxes de programación, orientadas á xestión de información. Co desenvolvemento dos editores e procesadores de texto xorden as primeiras linguaxes informáticas especializadas en tarefas de descrición e estruturación de información: as linguaxes de marcas. Paralelamente, tamén xorden outras linguaxes informáticas orientadas á representación, almacenamento e consulta 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 texto introducen nos documentos para dirixir o proceso de presentación (impresión) mediante unha 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, neles, inicialmente non había nada que lle permitise ao programador (formateador de documentos neste caso) abstraerse das características do procesador de textos e expresar de forma independente a este a estrutura e a lóxica interna do documento.



### Exemplo

Código de marcas anterior a GML. As etiquetas son de invención propia.

Dado o seguinte documento:

```
<times 14><cor verde><centrado> Este texto é un exemplo para mostrar a utilización primitiva das marcas</centrado>
</cor></times 14>
```

```
<cor granate><times 10><cursiva>Para realizar este exemplo utilízanse etiquetas da nosa invención. </cursiva> As
partes importantes do texto poden resaltarse usando a <negra>negra</negra>, ou o <subliñar>subliñado</subliñar>
</times 10></cor>
```

Ao imprimilo obteríase:

Este texto é un exemplo para mostrar a utilización primitiva das marcas

*Para realizar este exemplo utilízanse etiquetas da nosa invención. As partes importantes do texto poden resaltarse usando a **negra** ou o subliñado*

Posteriormente, engadíronse como medio de presentación á pantalla. Os códigos de estilo de visualización anteriores xa non aparecen, e empréganse outros medios para marcados, distintos da inclusión a man de cadeas formateadoras, agora ese proceso automatízase e basta premer unha combinación de teclas, ou pulsar un botón, para lograr os resultados requiridos. Aínda que isto é só unha abstracción, para o seu uso interno as aplicacións seguen utilizando marcas para delimitar aquelas partes do texto que teñen un formato especial.

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

## 2.1 GML (Generalizaed Markup Language).

---

Un dos problemas que se coñecen desde hai décadas na informática é a falta de estandarización nos formatos de información usados polos distintos programas.

Para resolver este problema, nos anos sesenta IBM encargoulle a Charles F. Goldfab a construción dun sistema de edición, almacenamento e procura de documentos legais. Tras analizar o funcionamento da empresa chegaron á conclusión de que para realizar un bo procesado informático dos documentos había que establecer un formato estándar para todos os documentos que se manexaban na empresa. Con iso lográbase xestionar calquera documento en calquera departamento e con calquera aplicación, sen ter en conta onde nin con que se xerou o documento. O dito formato tiña que ser válido para os distintos tipos de documentos legais que utilizaba a empresa, polo tanto, debía ser flexible para que se puidese axustar ás distintas situacións.

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

## 2.2 SGML (Standard Generalized Markup Language)

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



### Exemplo

Documento SGML sinxelo:

`<email>`

`<remitente>`

`<persoa>`

`<nome> Pepe </nome>`

`<apelido> Pérez</apelido>`

`</persoa>`

`</remitente>`

`<destinatario>`

`<endereço> pinocho@hotmail.com </endereço>`

`</destinatario>`

`<asunto> quedamos? </asunto>`

`<mensaxe> Ola, vin que poñen esta noite a película que querías ver. Apetéceche ir? </mensaxe>`

`</correo electrónico>`

## 2.3 HTML (HyperText Markup Language)

En 1989/90 Tim Berners-Le creou o World Wide Web e atopouse coa necesidade de organizar, enlazar e compatibilizar gran cantidade de información procedente de diversos sistemas. Para resolvelo creou unha linguaxe de descrición de documentos chamado HTML, que, en realidade, era unha combinación de dous estándares xa existentes:

- **ASCII**: é o formato que calquera procesador de textos sinxelo pode recoñecer e almacenar. Polo tanto é un formato que permite a transferencia de datos entre diferentes computadores.
- **SGML**: linguaxe que permite dar estrutura ao texto, resaltando os títulos ou aplicando diversos formatos ao texto.

HTML é unha versión simplificada de SGML, xa que só se utilizaban as instrucións absolutamente imprescindibles. Era tan fácil de comprender que rapidamente tivo grande aceptación logrando o que non puido SGML, HTML converteuse nun estándar xeral para a creación de páxinas web. Ademais, tanto as ferramentas de software como os navegadores que permiten visualizar páxinas HTML son cada vez mellores.

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

- Non soporta tarefas de impresión e deseño.
- A linguaxe non é flexible, xa que as etiquetas son limitadas.
- Non permite mostrar contido dinámico.
- A estrutura e o deseño están mesturados no documento.



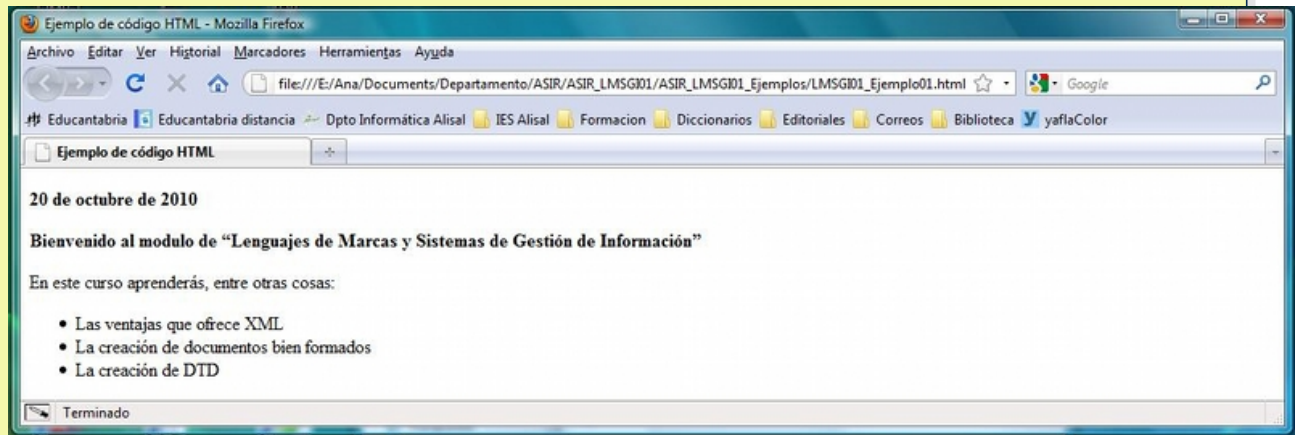
### Exemplo

Documento HTML

```
<html>
  <head>
    <title> Exemplo de código HTML</title>
  </head>
  <body bgcolor="#ffffff">
    <p></p>
    <p>
      <b>20 de outubro de 2010</b>
    </p>
    <p><b> Benvido ao modulo de "Linguaxes de Marcas e Sistemas de Xestión de Información"
    </b></p>
    <p> Neste curso aprender&aacute;s, entre outras cousas: <br/>
    <ul>
      <li>As vantaxes 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>

Ao publicalo nun navegador, por exemplo no Firefox, teriamos:



Captura de pantalla de Mozilla Firefox. Uso educativo non comercial. Elaboración propia



## 2.4 XML (extensible Markup Language)

Para resolver estes problemas de HTML o W3C establece, en 1998, o estándar internacional XML, unha linguaxe de marcas puramente estrutural que **non inclúe ningunha información relativa ao deseño**. Está a se converter con rapidez en estándar para o intercambio de datos na web. A diferenza de HTML as etiquetas indican o significado dos datos en lugar do formato co que se van a visualizar os datos.

XML é unha **metalinguaxe** caracterizada por:

- Permitir definir etiquetas propias.
- Permitir asignar atributos ás etiquetas.
- Utilizar un esquema para definir de forma exacta as etiquetas e os atributos.
- A estrutura e o deseño son independentes.

En realidade XML é un conxunto de estándares relacionados entre si e que son:

- **XSL**, extensible Style Language. Permite definir follas de estilo para os 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**. Provéen un contexto ao que se aplican as marcas dun documento de XML e que serve para diferencialas doutras con idéntico nome válidas noutros contextos.
- **XML Schemas**. Permiten definir restricións que se aplicarán a un documento XML. Actualmente os máis usados son as DTD.

En realidade XML é un conxunto de estándares relacionados entre si e que son:



### Exercicio resolto

Documento XML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE biblioteca">
<biblioteca>
  <exemplar tipo_ejem="libro" titulo="XML practico" editorial="Edicións Eni">
    <tipo> <libro isbn="978-2-7460-4958-1" edicion="1" paginas="347"></libro> </tipo>
    <autor nomee="Sebastien Lecomte"></autor>
    <autor nomee="Thierry Boulanger"></autor>
    <autor nomee="Anxo Belinchon Quella" funcion="tradutor"></autor>
    <prestado lector="Pepe Pérez">
      <data_pres dia="13" mes="mar" ano="2009"></data_pres>
      <data_devol dia="21" mes="jun" ano="2009"></data_devol>
    </prestado>
  </exemplar>
  <exemplar tipo_ejem="revista" titulo="Todo Linux 101. Virtualización en GNU/Linux" editorial="Studio Press">
```

```
<tipo>
  <revista>
    <data_publicacion mes="abr" ano="2009"></data_publicacion>
  </revista>
</tipo>
<autor nomee="Varios"></autor>
<prestado lector="Pedro Picapiedra">
  <data_pres dia="12" mes="ene" ano="2010"></data_pres>
</prestado>
</exemplar>
</biblioteca>
```

## 2.5 Comparación de XML con HTML

### XML

- É un perfil de SGML.
- Especifica como deben definirse conxuntos de etiquetas aplicables a un tipo de documento.
- Modelo de hiperenlaces complexo.
- O navegador é unha plataforma para o desenvolvemento de aplicacións.
- Fin da guerra dos navegadores e etiquetas propietarias.

### HTML

- É unha aplicación de SGML.
- Aplica un conxunto limitado de etiquetas sobre un único tipo de documento.
- Modelo de hiperenlaces simple.
- O navegador é un visor de páxinas.
- O problema da "non compatibilidade" e as diferenzas entre navegadores alcanzou un punto no que a solución é difícil

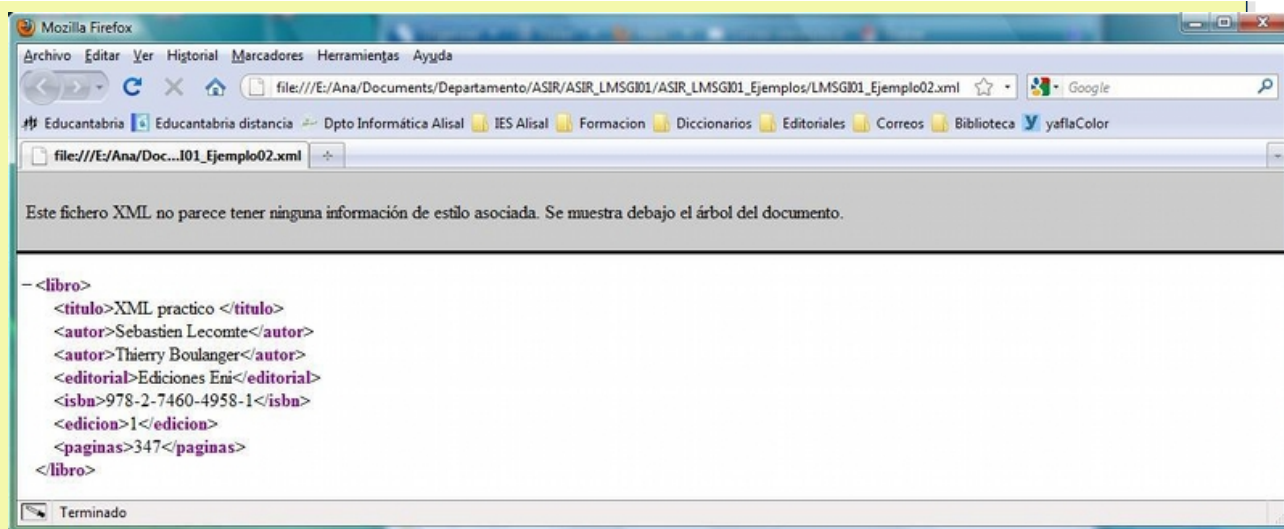


### Exemplo

#### Ficheiro XML

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

Ao interpretar este ficheiro cun navegador, por exemplo Mozilla, obtense:



Captura de pantalla Mozilla Firefox. Uso educativo non comercial. Elaboración propia

## Ficheiro HTML

```
<html>

  <head>

    <title>Libro</title>

  </head>

  <body>

    <h3>XML practico</h3><br>

    <p>autores: Sebastien Lecomte,
    Thierry Boulanger</p>

    <ul>

      <li>editorial: Edicións Eni</li>

      <li>isbn:978-2-7460-4958-1</li>

      <li>edicion: 1 </li>

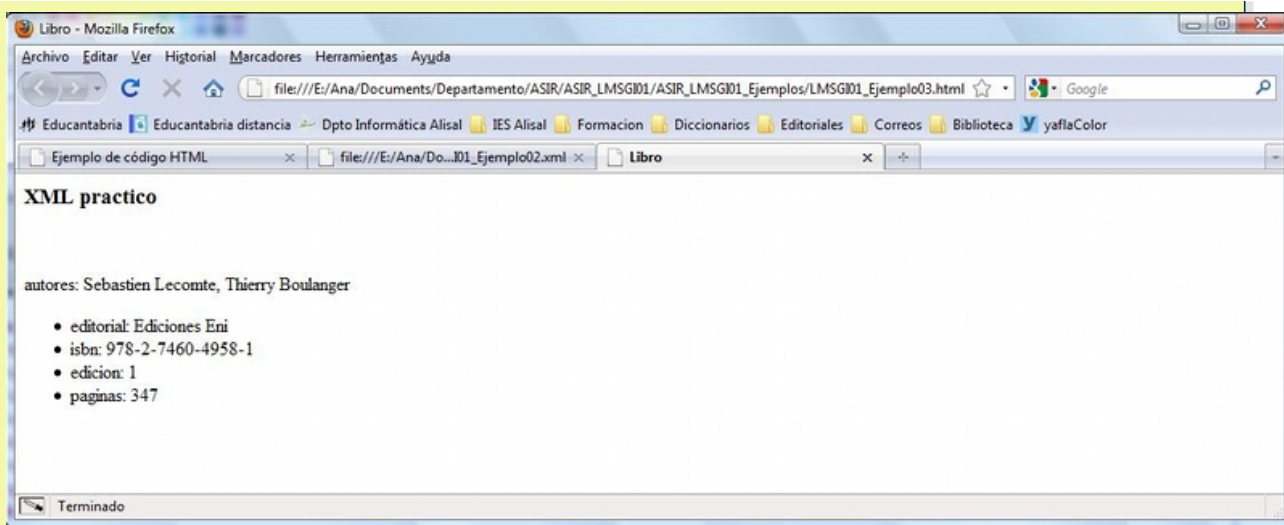
      <li>paginas: 347</li>

    </ul>

  </body>

</html>
```

Ao interpretalo co navegador Mozilla Firefox teremos:



Captura de pantalla Mozilla Firefox. Uso educativo non comercial. Elaboración Propia

## 2.6. Comparación de XML con SGML

### XML

- O seu uso é sinxelo.
- Traballa con documentos **ben formados**, non esixe que estean validados.
- Facilita o desenvolvemento de aplicacións de baixo custo.
- É moi utilizado en informática e en máis áreas de aplicación.
- Compatibilidade e integración con HTML.
- Formateo e estilos fáciles de aplicar.
- Non usa etiquetas opcionais.

### SGML

- O seu uso é moi complexo.
- Só traballa con documentos **válidos**.
- A súa complexidade fai que as aplicacións informáticas para procesar SGML sexan moi custosas.
- Só se utiliza en sectores moi específicos.
- Non hai unha compatibilidade con HTML definida.
- Formateo e estilos relativamente complexos.



### Pregunta de elección múltiple

Cales son as características comúns de XML e SGML?

- ☐ a) Gardar o formato dun documento.
- ☐ b) Gardar a estrutura lóxica dos documentos.
- ☐ c) Gardar os documentos no formato universal txt.
- ☐ d) Gardar o formato dos documentos independentemente da plataforma.



### Para saber máis

A recomendación de XML publicada polo **W3C** é pública e accesible en:

<http://www.w3.org/TR/2000/REC-xml-20001006>

### 3. Etiquetas

As linguaxes de marcas utilizan unha serie de etiquetas especiais intercaladas nun documento de texto sen formato. As ditas etiquetas serán posteriormente interpretadas polos intérpretes da linguaxe e axudan ao procesado do documento.

As etiquetas **escribíense encerradas entre ángulos**, é dicir `<e>`. Normalmente, **utilízanse dúas etiquetas: unha de inicio e outra de fin** para indicar que terminou o efecto que queríamos presentar. A única diferenza entre ambas é que a de peche leva unha barra inclinada "/" antes do código.

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



#### Exemplo

Por exemplo, en HTML

`<u>`Isto está subliñado`</u>`

Ao interpretalo nun navegador verase así:

Isto está subliñado

As últimas especificacións emitidas por [o W3C](#) indican a necesidade de que vaian escritas **sempre en minúsculas** para considerar que o documento está correctamente creado.



#### Autoavaliación

Cal das seguintes liñas é correcta?

- ☐ a) `<i>`Texto en cursiva
- ☐ b) `<i>`Texto en cursiva`<i>`
- ☐ c) `<i>`Texto en cursiva`</i>`
- ☐ d) `<l>`Texto en cursiva`<l>`

## 4. Ferramentas básicas de XML

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

### ◦ Editores XML

Unha característica das linguaxes de marcas é que se basean na utilización de ficheiros de texto **plano** polo que basta utilizar un procesador de texto normal e corrente para construír un documento XML.

Para crear documentos XML complexos e ir engadindo datos é conveniente usar algún editor XML. Estes axúdannos a crear estruturas e etiquetas dos elementos usados nos documentos, ademais algúns inclúen axuda para a creación doutros elementos como DTD, follas de estilo CSS ou XSL... O **W3C** desenvolveu un editor de HTML, XHTML, CSS e XML gratuíto cuxo nome é Amaia.

### ◦ Procesadores XML

Para interpretar o código XML pódese utilizar calquera navegador. Os procesadores de XML permiten ler os 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 de XML que comproba que o documento cumpre as normas establecidas para que poida abrirse. Estas normas poden corresponderse coas necesarias para traballar só con documentos **de tipo válido** ou só esixir que o **documento estea ben formado**, os primeiros coñécense como validadores e os segundos como non validadores. O modo en que os procesadores deben ler os datos XML está descrito na recomendación de XML establecida por W3C .

Para publicar un documento XML en internet utilízanse os procesadores XSLT, que permiten xerar arquivos HTML a partir de documentos XML.

Posto que XML se pode utilizar para o intercambio de datos entre aplicacións, hai que recorrer a motores independentes que se executan sen que nos deamos conta. Entre estes destacan "XML para Java" de IBM, JAXP de Sun etc.



### Autoavaliación

Para crear documentos XML é necesario:

- ☐ a) Software especializado para a tecnoloxía XML.
- ☐ b) Ferramentas de validación de XML.
- ☐ c) Un block de notas e un navegador.
- ☐ d) Polo menos, un editor XML.



### Para saber máis

Información sobre analizadores XML:

<http://www.oasis-open.org/cover>

Algúns dos analizadores dispoñibles están nas ligazóns seguintes:

<http://www.jclark.com/xml/expat.html>

<http://www.mozilla.org/rdf/doc/xml.html>





## 5. XML: estrutura e sintaxe

---

O XML, ou Linguaxe de Etiquetas Estendida, é linguaxe de etiquetas, creadas polo programador, que estruturan e gardan de forma ordenada a información. Non representa datos en por si, soamente organiza a estrutura.

O XML aforra tempos de desenvolvemento e proporciona vantaxes, dotando a webs e a aplicacións dunha forma realmente potente de gardar a información. Ademais, converteuse nun formato universal que foi asimilado por todo tipo de sistemas operativos e dispositivos móbiles.

Do mesmo xeito que en HTML un documento XML é un documento de texto, neste caso con extensión ".xml", composto de parellas de etiquetas, estruturadas en árbore, que describen unha función na organización do documento, que pode editarse con calquera editor de texto e que é interpretado pola navegadores web.

As características básicas de XML son:

- Dado que XML concibiuse para traballar na web, é directamente compatible con protocolos que xa funcionan, como HTTP e os URL.
- Todo documento que verifique as regras de XML está conforme con SGML.
- Non se requiren coñecementos de programación para realizar tarefas sinxelas en XML.
- Os documentos XML son fáciles de crear.
- A difusión dos documentos XML está asegurada xa que calquera procesador de XML pode ler un documento de XML.
- O marcado de XML é lexible para os humanos.
- O deseño XML é formal e conciso.
- XML é extensible, adaptable e aplicable a unha gran variedade de situacións.
- XML é orientado a obxectos.
- Todo documento XML se compón exclusivamente de datos de marcado e datos carácter mesturados.

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

- Especificación de requisitos.
- Deseño de etiquetas.
- Marcado dos documentos.

O marcado en XML son etiquetas que se engaden a un texto para estruturar o contido do documento. Esta información extra permítelles aos computadores "interpretar" os textos. O marcado é todo o que se sitúa entre os caracteres "<" e ">" o "&" y ";"

Os datos 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 interprete XML. Estes inclúense entre as cadeas "<!--" y "-->", poden estar en calquera posición no documento salvo:

- Antes do prólogo.

- Dentro dunha etiqueta.

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

## 5.1. O prólogo

Se se inclúe, o prólogo **debe preceder o exemplar do documento**. A súa inclusión facilita o procesado da información do exemplar. O prólogo está dividido en dúas partes:

- **A declaración XML:** no caso de incluírse ha de ser a primeira liña do documento, de non ser así se xera un erro que impide que o documento sexa procesado.

O feito de que sexa opcional permite o procesamento de documentos HTML e SGML coma se fosen XML, se fose obrigatoria estes deberían incluír unha declaración de versión XML que non teñen.

O prólogo pode ter tres funcións:

**A declaración do tipo de documento**, define que tipo de documento estamos a crear para ser procesado correctamente. Toda declaración de tipo de documento comeza pola cadea:

- *Declaración a versión de XML usada para elaborar o documento.*

Para iso utilízase a etiqueta:

```
<?xml versión= "1.0" ?>
```

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

- *Declaración da codificación empregada para representar os caracteres.*

Determina o conxunto de caracteres que se utiliza no documento. Para iso escríbese:

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

Neste caso úsase o código iso-8859-1 (Latin-1) que permite o uso de acentos ou caracteres como a ñ.

Os códigos máis importantes son:

Estándar ISO	Código de país
UTF-8 (Unicode)	Conxunto de caracteres universal
ISO -8859-1 (Latín-1)	Europa occidental, Latinoamérica
ISO -8859-2 (Latín-2)	Europa central e oriental
ISO -8859-3 (Latín-3)	Suroeste de Europa
ISO -8859-4 (Latín-4)	Países Escandinavos, Bálticos
ISO -8859-5	Cirílico
ISO -8859-6	Árabe
ISO -8859-7	Grego
ISO -8859-8	Hebreo
ISO -8859-9	Turco

ISO-8859-10	Lapón. Nórdico, esquimó
EUC-JP oder Shift_JIS	Xaponés

◦ *Declaración da autonomía do documento.*

Informa de se o documento necesita doutro para a súa interpretación. Para declaralo hai que definir o prólogo completo:

`<?xml versión= "1.0" encoding="iso-8859-1" standalone="non" ?>`

Neste caso, o documento é independente, de non ser así o atributo standalone tomase o valor "yes".

- A declaración do tipo de documento, define que tipo de documento estamos creando para ser procesado correctamente. Toda declaración de tipo de documento comeza pola cadea:

`<!DOCTYPE Nomee_tipo ...>`

## 5.2 O exemplar. Os 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 distintos bloques de información que permiten definir 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 por atributos.



### Exemplo

Sexa o seguinte código XML

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

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

En realidade, **o exemplar é o elemento raíz** dun documento XML. **Todos os datos** dun documento XML **han de pertencer a un elemento** deste.

Os nomes das etiquetas han de ser autodescritivos, o que facilita o traballo que se fai con elas.

A formación de elementos ha de cumprir certas normas para que queden perfectamente definidos e que o documento XML ao que pertencen poida ser interpretado polos procesadores XML sen xerar ningún erro fatal. As ditas regras son:

- En todo documento XML debe existir un elemento raíz, e só un.
- Todos os elementos teñen unha etiqueta de inicio e outra de peche. No caso de que no documento existan elementos baleiros, pódense substituír as etiquetas de inicio e peche por unha de elemento baleiro. Esta constrúese como a etiqueta de inicio, pero substituíndo o carácter ">" por / ">". É dicir, `<elemento></elemento>` pode substituírse por: `<elemento/>`
- Ao aniñar elementos hai que ter en conta que non pode pecharse un elemento que conteña algún outro elemento que aínda non se pechou.
- Os nomes das etiquetas de inicio e de peche dun mesmo elemento han de ser idénticos, respectando as maiúsculas e minúsculas. Poden ser calquera cadea alfanumérica que non conteña espazos e non comece nin polo carácter dous puntos, ":", nin pola cadea "xml" nin ningunha das súas versións en que se cambien maiúsculas e minúsculas ("XML", "XML", "xml"...).

- O contido dos elementos non pode conter a cadea "]]>" por compatibilidade con SGML. Ademais non se poden utilizar directamente os caracteres maior que, >, menor que, <, ampersand, &, dobres comiñas, ", e apostrofe, '. No caso de ter que utilizar estes caracteres substitúense polas seguintes cadeas:

Carácter	Cadea
>	&gt;
<	&lt;

Carácter	Cadea
&	&amp;
"	&quot;

Carácter	Cadea
'	&apos;

- Para utilizar caracteres especiais, como £, ©, ®,... hai que usar as expresións &#D; ou &#H; onde D e H se corresponden respectivamente co número decimal ou hexadecimal correspondente ao carácter que se quere representar no código UNICODE. Por exemplo, para incluír o carácter de Euro, €, usaríanse as cadeas &#8364; ou &#x20AC;



## Debes coñecer

Na seguinte ligazón atoparás unha táboa cos caracteres [ASCII](http://ascii.cl/es/codigos-html.htm), o nome HTML, e o número HTML de cada un deles que che será imprescindible á hora de realizar documentos en HTML e XML.

<http://ascii.cl/es/codigos-html.htm>

## 5.2.1. Atributos

Permiten engadir propiedades aos elementos dun documento. Os atributos non poden organizarse en ningunha xerarquía, non poden conter ningún outro elemento ou atributo e non reflicten ningunha estrutura lóxica.

Non se debe utilizar un atributo para conter información susceptible de ser dividido.

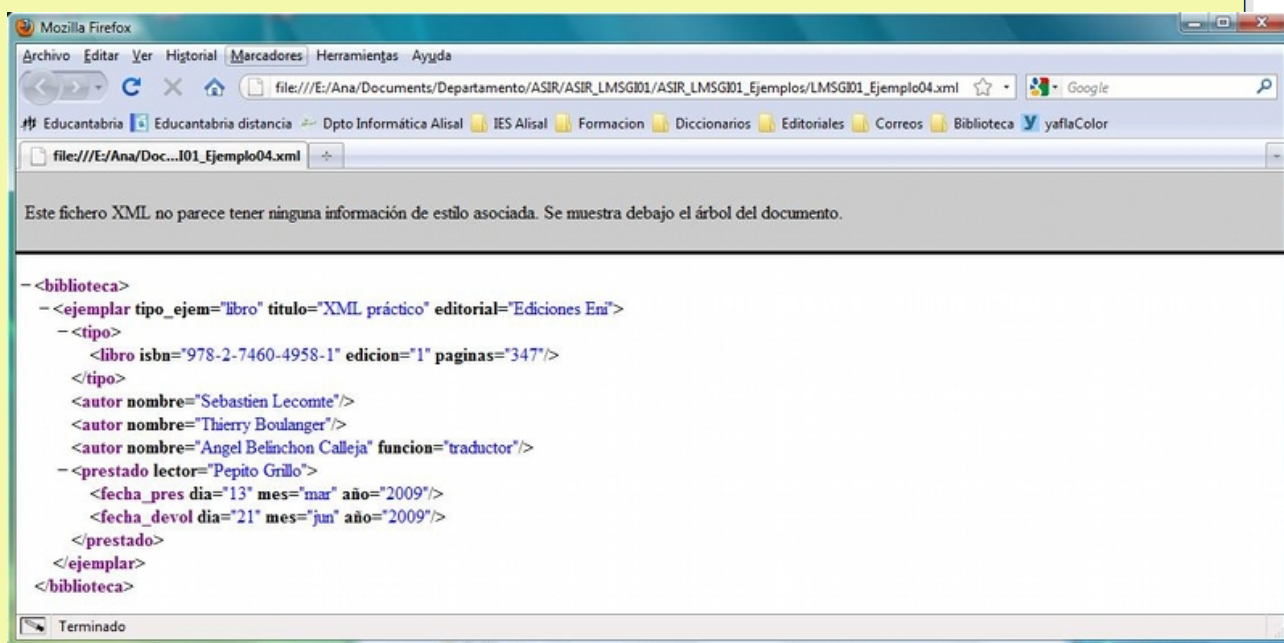


### Exemplo

Dado o seguinte código XML:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?
<!DOCTYPE biblioteca >
<biblioteca>
  <exemplar tipo_ejem="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="Anxo Belinchon Rúa" funcion="tradutor"></autor>
    <prestado lector="Pepe Pérezlo">
      <data_pres dia="13" mes="mar" ano="2009"></data_pres>
      <data_devol dia="21" mes="jun" ano="2009"></data_devol>
    </prestado>
  </exemplar>
</biblioteca>
```

Ao abrir o documento anterior co navegador Firefox obtemos:



Captura de pantalla de Mozilla Firefox. Uso educativo non comercial. Elaboración Propia

Vemos que os elementos aparecen coloreados en maxenta, os nomes dos atributos en negro e os seus valores en azul.



Como se observa no exemplo, os atributos defínense e dan valor dentro dunha etiqueta de inicio ou de elemento baleiro, a continuación do nome do elemento ou da definición doutro atributo sempre separado deles por un espazo. Os valores do atributo van precedidos dun igual que segue ao nome deste e teñen que definirse entre comiñas simples ou dobres.

Os nomes dos atributos han de cumprir as mesmas regras que os dos elementos, e non poden conter o carácter menor que, <.



## Autoavaliación

Cales son os erros do seguinte documento XML?

```
<?XML version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE biblioteca >
<biblioteca>
  <exemplar tipo Ejem="libro" titulo="XML práctico" editorial="Edicions Eni">
    <tipo> <libro isbn="978-2-7460-4958-1" edicion= paxinas="347"></libro> </tipo>
    <autor nome="Sebastien Lecomte"></autor>
    <autor nome="Thierry Boulanger"></autor>
    <autor nomee="Angel Belinchon Rúa" funcion="tradutor"></autor>
    <prestado lector="Pepe Pérez">
      <data_pres dia="13" mes="mar" ano="2009"></data_pres>
      <data_devol/>
    </prestado>
  </exemplar>
</biblioteca>
```

- ☐ a) Utiliza maiúsculas na definición da versión XML.
- ☐ b) Non utiliza o código de caracteres adecuado.
- ☐ c) Os valores dos atributos non están entre comiñas dobres.
- ☐ d) Hai algún atributo baleiro.
- ☐ e) A etiqueta <data\_devol/> non se pecha.
- ☐ f) Úsanse maiúsculas nos datos do documento.

## 6. Documentos XML ben formados

Todos os documentos XML deben verificar as regras sintácticas que define a recomendación do [W3C](#) para o estándar XML. Esas normas básicas son:

- O documento ha de ter definido un prólogo coa declaración xml completa.
- Existe un único elemento raíz para cada documento: é un só elemento no que todos os demais elementos e contidos se atopan aniñados.
- Hai que cumprir as regras sintácticas do linguaxe XML para definir os distintos elementos e atributos do documento



### Autoavaliación

Está "ben formado" o seguinte documento XML?

```
<?xml version="1.0"?>
<mensaxe>
  <destinataria>Tomas</ destinatario>
  <remitente>Xoán</ remitente>
  <asunto>
    <contido> Non esquezas ir recollerme ao aeroporto mañá pola mañá!</contido>
  </asunto>
</mensaxe>
```

☐ Verdadeiro ☐ Falso

## 7. Utilización de espazos de nomes en XML

Permiten definir a pertenza dos elementos e os atributos dun documento XML ao contexto dun vocabulario XML. Deste xeito resólvense as ambigüidades que se poden producir ao xuntar dous documentos distintos, de dous autores diferentes, que utilizaron o mesmo nome de etiqueta para representar cousas distintas.

Os espazos de nomes tamén coñecidos como name spaces, permiten dar un nome único a cada elemento, indexándoos segundo o nome do vocabulario adecuado ademais están asociados a un [URI](#) que os identifica de forma única.

No documento, as etiquetas ambiguas substitúense por outras nas que o nome do elemento está 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 denomínase "nome cualificado". Ao definir o prefixo hai que ter en conta que non se poden utilizar espazos nin caracteres espaciais e que non pode comezar por un dígito.

Antes de poder utilizar un prefixo dun espazo de nomes, para resolver a ambigüidade de dúas ou máis etiquetas, é necesario declarar o espazo de nomes, é dicir, asociar un índice co [URI](#) asignado ao espazo de nomes, mediante un atributo especial `xmlns`. Isto faise entre o prólogo e o exemplar dun documento XML e a súa sintaxe é a seguinte:

```
<conexion>://<enderezoservidor>/<apartado1>/<apartado2>/...
```



### Exemplo

Sexan os documentos XML que organizan a información sobre os profesores e os alumnos do ASIR respectivamente:

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

```
<!DOCTYPE alumnos>
```

```
<alumnos>
```

```
  <nome>Fernando Fernández González</nome>
```

```
  <nome>Isabel González Fernández</nome>
```

```
  <nome>Ricardo Martínez López</nome>
```

```
</alumnos>
```

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

```
<!DOCTYPE profesores>
```

```
<profesores>
```

```
  <nome>Pilar Ruíz Pérez</nome>
```

```
  <nome>Tomás Rodríguez Hernández</nome>
```

```
</profesores>
```

Ao facer un documento sobre os membros do curso ASIR non se distinguirían os profesores dos alumnos, para resolvelo definiremos un espazo de nomes para cada contexto:

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

<!DOCTYPE membros>

<alumnos xmlns:alumnos="http://ASIR/alumnos">

<profesores xmlns:profesores="http://ASIR/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>
```



## Autoavaliación

Os espazos de nomes permiten

- ☐ a) Utilizar etiquetas idénticas para estruturar distintos tipos de información de texto.
- ☐ b) Estructurar a información dun documento XML cando provén de varios documentos.
- ☐ c) Asignar varias etiquetas a unha mesma información.
- ☐ d) Definir etiquetas noutros documentos.



## Para saber máis

Os espazos de nomes teñen unha recomendación en XML:

<http://www.w3.org/TR/REC-xml-names/>