UT 1. Introducción a los Lenguajes de Marcas

LENGUAJES DE MARCAS Y SISTEMAS DE GESTIÓN DE INFORMACIÓN

I.E.S. Luis Vives – Desarrollo de Aplicaciones Web

Conceptos

Un *lenguaje de marcas* es un modo de codificar un documento donde, junto con el texto, se incorporan etiquetas, marcas o anotaciones con información adicional relativa a la estructura del texto o su formato de presentación. Permiten hacer explícita la estructura de un documento, su contenido semántico o cualquier otra información lingüística o extralingüística que se quiera hacer patente.

Todo lenguaje de marcas está definido en un documento denominado DTD (Document Type Definition). En él se establecen las marcas, los elementos utilizados por dicho lenguaje y sus correspondientes etiquetas y atributos, su sintaxis y normas de uso.

Ejemplo

```
<noticia>
   <lugar>Madrid</lugar>
   <fecha>27/08/2010</fecha>
   <desc>Se ha inaugurado una estación de tren</desc>
</noticia>
```

Tipos de lenguajes de marca

Aunque en la práctica, en un mismo documento pueden combinarse varios tipos diferentes de lenguajes de marca, éstos se pueden clasificar como sigue:

- De presentación
- De procedimientos
- Descriptivo o semántico

Son los **usados tradicionalmente por los procesadores de texto** (como puede ser Microsoft Word) y codifican cómo ha de presentarse el documento, *por ejemplo, indicando que una determinada palabra debe presentarse en fuente itálica o que debe dejar un espacio de 10 puntos al terminar el párrafo.*

Generalmente **las marcas de los lenguajes orientados a presentación se ocultan al usuario** lo que permite obtener un efecto WYSIWYG (What you see is what you get).

Este tipo de lenguajes de marcas no suelen ser flexibles ni reusables.

Ejemplo LdM de presentación : rtf

Orientado también a la presentación pero, en este caso, **el programa que representa el documento debe interpretar el código en el mismo orden en que aparece**.

Ejemplos: TeX, LaTeX y Postcript

Latex:

Instalación. : Miktex. Usar la edición portable. Una vez instalado, ir al directorio de instalación y doble click sobre miktex_portable.cmd. Después, en iconos de notificación pulsar con botón dcho sobre el de miktex ->TexWorks y copia el fichero (o teclea) .tex sobre el área de trabajo. Pulsa Compilar ->Compilar y espera un ratito aceptando la instalación de los diferentes paquetes necesarios. Si todo va bien, en el directorio de miktex tendremos el pdf generado.

Uso básico de latex.

LdM descriptivo o semántico

Este tipo de lenguajes no definen qué se debe hacer con un trozo o sección del documento sino que por el contrario **las marcas sirven para indicar qué es esa información**, esto es, describen que es lo que se está representando.

La mayoría de los lenguajes de marcas que se usan hoy en día se encuentran dentro de este grupo, como SGML y sus derivados (HTML, XML, etc..)

Ejemplo

```
<html>
<head>
</head>
<body>
<ol>
  <li>Fruta</li>
    <ul type="circle">
      <li>Manzana </li>
      <li>Naranja </li>
    </ul>
  <li>Verdura </li>
    <ul type="disc">
      <li>Zanahoria </li>
      <li>Lechuga </li>
    </ul>
</ol>
</body>
</html>
```

1. Fruta
   - Manzana
   - Naranja
2. Verdura
   - Zanahoria
   - Lechuga

La marca <ol> indica que la información que se va a representar es una lista ordenada

Ejemplos de usos del lenguaje de marcas

Documentación electrónica

RTF: (Rich Text Format) Formato de Texto Enriquecido, fue desarrollado por Microsoft en 1987. Permite el intercambio de documentos de texto ente distintos procesadores de texto.

TeX: Su objetivo es la creación de ecuaciones matemáticas complejas.

Wikitexto: Permite la creación de páginas wiki en servidores preparados para soportar este lenguaje.

Tecnologías de internet

HTML, XHTML: (Hypertext Markup Language, eXtensible Hypertext Markup Language): Su objetivo es la creación de páginas web.

RSS: Permite la difusión de contenidos web

Otros lenguajes especializados

MathML (Mathematical Markup Language): Su objetivo es expresar el formalismo matemático de tal modo que pueda ser entendido por distintos sistemas y aplicaciones.

VoiceXML (Voice Extended Markup Language) tiene como objetivo el intercambio de información entre un usuario y una aplicación con capacidad de reconocimiento de habla.

MusicXML: Permite el intercambio de partituras entre distintos editores de partituras.

Evolución

Evolución

En los años 70 surgen unos lenguajes informáticos, distintos de los lenguajes de programación, orientados a la gestión de información. Con el desarrollo de los editores y procesadores de texto surgen los primeros lenguajes informáticos especializados en tareas de descripción y estructuración de información: los lenguajes de marcas.

Los lenguajes de marcas surgieron, inicialmente, como lenguajes formados por el conjunto de códigos de formato que los procesadores de texto introducen en los documentos para dirigir el proceso de presentación (impresión) mediante una impresora. Como en el caso de los lenguajes de programación, inicialmente estos códigos de formato estaban ligados a las características de una máquina, programa o procesador de textos concreto y, en ellos, inicialmente no había nada que permitiese al programador (formateador de documentos en este caso) abstraerse de las características del procesador de textos y expresar de forma independiente a éste la estructura y la lógica interna del documento.

Este marcado estaba exclusivamente orientado a la presentación de la información, aunque pronto se percataron de las posibilidades del marcado y le dieron nuevos usos que resolvían una gran variedad de necesidades, apareció el formato generalizado.

Generalized Markup Language

GML

Uno de los problemas que se conocen desde hace décadas en la informática es la falta de estandarización en los formatos de información usados por los distintos programas.

Para resolver este problema, en los años sesenta IBM encargó a Charles F. Goldfab la construcción de un sistema de edición, almacenamiento y búsqueda de documentos legales. Tras analizar el funcionamiento de la empresa llegaron a la conclusión de que para realizar un buen procesado informático de los documentos había que establecer un formato estándar para todos los documentos que se manejaban en la empresa. Con ello se lograba gestionar cualquier documento en cualquier departamento y con cualquier aplicación, sin tener en cuenta dónde ni con qué se generó el documento. Dicho formato tenía que ser válido para los distintos tipos de documentos legales que utilizaba la empresa, por tanto, debía ser flexible para que se pudiera ajustar a las distintas situaciones.

El formato de documentos que se creó como resultado de este trabajo fue GML, cuyo objetivo era describir los documentos de tal modo que el resultado fuese independiente de la plataforma y la aplicación utilizada.

Standard Generalized Markup Language

SGML

El formato GML evolucionó hasta que en 1986 dio lugar al estándar ISO 8879 que se denominó SGML.

Este estándar tuvo una gran aceptación pero no consiguió asentarse del todo debido principalmente a su complejidad lo que provocaba que el software que usará SGML terminaba siendo excesivamente extenso, complejo y requería de unas herramientas de software caras. Por ello su uso ha quedado relegado a grandes aplicaciones industriales.

HyperText Markup Language

HTML

En 1989/90 Tim Berners-Lee creó el **World Wide Web (W3C)** y se encontró con la necesidad de organizar, enlazar y compatibilizar gran cantidad de información procedente de diversos sistemas. Para resolverlo creó un lenguaje de descripción de documentos llamado HTML, que, en realidad, era una combinación de dos estándares ya existentes:

ASCII: Es el formato que cualquier procesador de textos sencillo puede reconocer y almacenar. Por tanto es un formato que permite la trasferencia de datos entre diferentes ordenadores.

SGML: Lenguaje que permite dar estructura al texto, resaltando los títulos o aplicando diversos formatos al texto.

HTML es una versión simplificada de *SGML*, ya que sólo se utilizaban las instrucciones absolutamente imprescindibles. Era tan fácil de comprender que rápidamente tuvo gran aceptación logrando lo que no pudo SGML, HTML se convirtió en un estándar general para la creación de páginas web. Además, tanto las herramientas de software como los navegadores que permiten visualizar páginas HTML son cada vez mejores. A pesar de todas estas ventajas HTML no es un lenguaje perfecto, sus principales desventajas son:

- El lenguaje no es flexible, ya que las etiquetas son limitadas.
- No permite mostrar contenido dinámico.
- La estructura y el diseño están mezclados en el documento.

Investiga sobre:

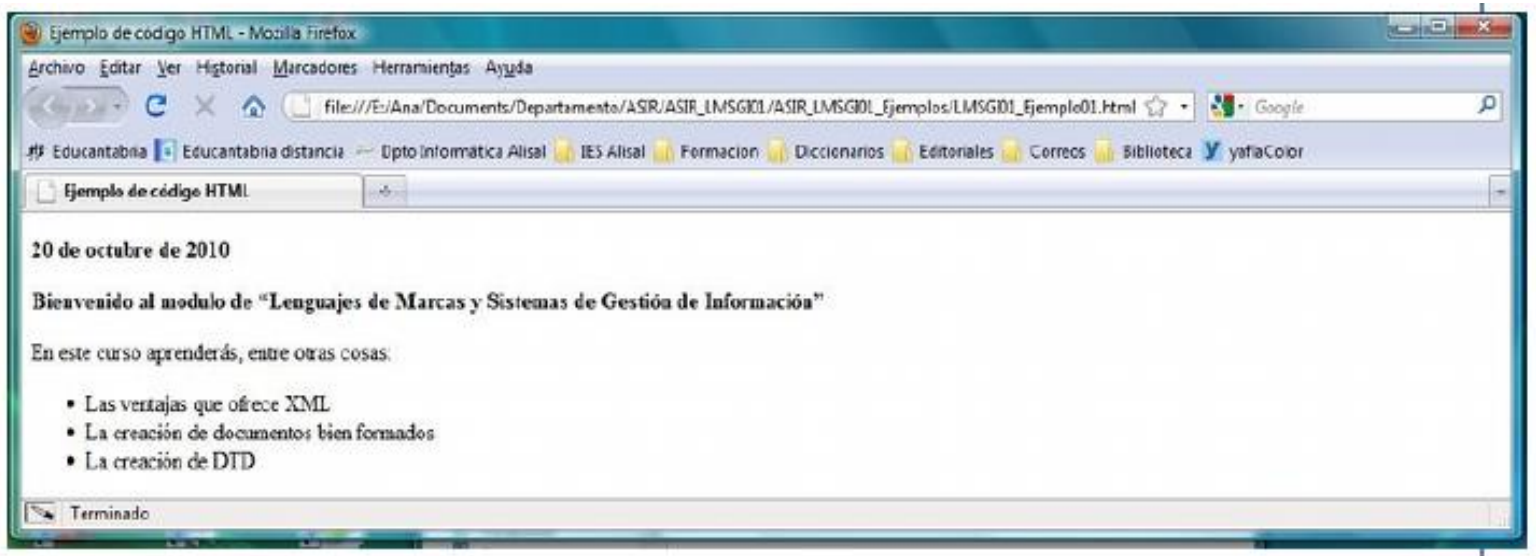
- ¿Qué es la World Wide Web?

Diferencia entre Internet y Web (WWW)

- **Internet es la red de conexiones entre servidores, World Wide Web es el sistema de navegación de páginas electrónicas.**
- Por ejemplo, cuando uno usa un programa para leer y enviar correo electrónico como Outlook lo que está haciendo es enviar y recibir mensajes en Internet, pero no está haciendo uso de la World Wide Web.
- Básicamente uno ocupa la World Wide Web cada vez que usa un navegador.
- De hecho, Internet existe antes que la World Wide Web, pero como los protocolos de comunicación eran tan sofisticados no se había masificado. Lo curioso es que la World Wide Web salió de uno de los institutos científicos más prestigiosos del mundo, el CERN. El CERN en Europa es el acelerador de partículas más grande del mundo. Allí trabajan científicos de todas las nacionalidades y el manejo de información siempre ha sido un problema.
- Por esta razón uno de estos científicos creo una forma de manejar la información de manera mas amigable. Básicamente creo un sistema para clasificar la información mediante páginas electrónicas y un navegador para ver las páginas. El tipo lo presento en una conferencia en CERN, pero allí encontraron que el sistema era muy simple y que poco podía aportar a la investigación científica. En 1993 CERN anuncio que el sistema World Wide Web seria gratis para todo el mundo, lo cual popularizo su uso. Así que gracias a ellos hoy en día cualquiera puede usar Internet mediante la World Wide Web de manera amigable sin tener que ser un experto en protocolos de comunicación.

Ejemplo

```
<html>
    <head>
        <title> Ejemplo de código HTML</title>
    </head>
    <body bgcolor="#ffffff">
        <p></p>
        <p>
            <b>20 de octubre de 2010</b>
        </p>
        <p><b> Bienvenido al modulo de “Lenguajes de Marcas y Sistemas de Gestión de Información”</b></p>
        <p> En este curso aprender&aacute;s, entre otras cosas:<br/>
        <ul>
            <li>Las ventajas que ofrece XML </li>
            <li>La creaci&oacute;n de documentos bien formados </li>
            <li>La creaci&oacute;n de DTD</li>
        </ul>
        </p>
    </body>
</html>
```



Ejemplo

Por ejemplo en HTML:

<u>Esto está subrayado</u>

Al interpretarlo el navegador:

<u>Esto está subrayado</u>

Ejercicio de clase.

¿Cuál de las siguientes líneas es correcta?

a) Texto en cursiva
b) Texto en cursiva
c) Texto en cursiva
d) <I>Texto en cursiva <I>

eXtensible Markup Language

XML

Para resolver estos problemas de HTML el W3C establece, en 1998, el estándar internacional XML, un lenguaje de marcas puramente estructural que no incluye ninguna información relativa al diseño. Está convirtiéndose con rapidez en estándar para el intercambio de datos en la Web. A diferencia de HTML las etiquetas indican el significado de los datos en lugar del formato con el que se van a visualizar los datos.

XML es un **metalenguaje** caracterizado por:

- Permitir definir etiquetas propias.
- Permitir asignar atributos a las etiquetas.
- Utilizar un esquema para definir de forma exacta las etiquetas y los atributos.
- La estructura y el diseño son independientes.

En realidad XML es un conjunto de estándares relacionados entre sí y que son:

- XSL, eXtensible Style Language. Permite definir hojas de estilo para los documentos XML e incluye capacidad para la transformación de documentos.
- XML Linking Language, incluye Xpath, Xlink y Xpointer. Determinan aspectos sobre los enlaces entre documentos XML.
- XML Namespaces. Proveen un contexto al que se aplican las marcas de un documento de XML y que sirve para diferenciarlas de otras con idéntico nombre válidas en otros contextos.
- definir restricciones que se aplicarán a un documento XML. Actualmente los más usados son las DTD. XML Schemas. Permiten

El XML, o Lenguaje de Etiquetas Extendido, es lenguaje de etiquetas, creadas por el programador, que estructuran y guardan de forma ordenada la información. No representa datos por sí mismo, solamente organiza la estructura.

El XML ahorra tiempos de desarrollo y proporciona ventajas, dotando a webs y a aplicaciones de una forma realmente potente de guardar la información. Además, se ha convertido en un formato universal que ha sido asimilado por todo tipo de sistemas operativos y dispositivos móviles.

Al igual que en HTML un documento XML es un documento de texto, en este caso con extensión *".xml"*, compuesto de parejas de etiquetas, estructuradas en árbol, que describen una función en la organización del documento, que puede editarse con cualquier editor de texto y que es interpretado por los navegadores Web.

XML: Características básicas

- Dado que XML se concibió para trabajar en la Web, es directamente **compatible** con **protocolos** que ya funcionan, como **HTTP** y con las URL.
- **Todo documento** que verifique las reglas de **XML** está **conforme con SGML**.
- **No** se **requieren conocimientos de programación** para realizar tareas sencillas en XML.
- Los documentos XML son **fáciles de crear**.
- La **difusión de los documentos XML está asegurada** ya que cualquier procesador de XML puede leer un documento de XML.
- El **marcado de XML es legible** para los humanos.
- El **diseño XML es formal y conciso**.
- XML **es extensible, adaptable y aplicable** a una gran variedad de situaciones.
- XML es orientado a objetos.
- Todo documento XML se compone exclusivamente de datos de marcado y datos carácter entremezclados.

Proceso de creación de un XML

El proceso de creación de un documento XML pasa por varias etapas en las que el éxito de cada una de ellas se basa en la calidad de la anterior. Estas **etapas** son:

- **Especificación de requisitos**. (Análisis)
- **Diseño de etiquetas**. (Diseño)
- **Marcado de los documentos**. (Implementación)

Los documentos XML **pueden tener comentarios**, que no son interpretados por el interprete XML. Estos se incluyen entre las cadenas "<!-" y "-->", pueden estar en cualquier posición en el documento salvo:

- Antes del prólogo.
- Dentro de una etiqueta.

Los documentos XML pueden estar formados por una parte opcional llamada **prólogo** y otra parte obligatoria llamada **ejemplar**.

Etiquetas

Los lenguajes de marcas utilizan una serie de **etiquetas especiales intercaladas en un documento de texto sin formato**. Dichas etiquetas serán posteriormente interpretadas por los intérpretes del lenguaje y ayudan al procesado del documento.

Las etiquetas **se escriben encerradas entre ángulos**, es decir < y >. Normalmente, **se utilizan dos etiquetas: una de inicio y otra de fin** para indicar que ha terminado el efecto que queríamos presentar. La única diferencia entre ambas es que **la de cierre lleva una barra inclinada "/“ antes del código**.

<etiqueta> texto que sufrirá las consecuencias de la etiqueta </etiqueta>

Las últimas especificaciones emitidas por el W3C indican la necesidad de que vayan escritas **siempre en minúsculas** para considerar que el documento está correctamente creado.

Representan estructuras mediante las que se organizará el contenido del documento o acciones que se desencadenan cuando el programa navegador interpreta el documento.

Constan de la etiqueta de inicio, la etiqueta de fin y todo aquello que se encuentra entre ambas.

Atributos

Es un par **nombre-valor que se encuentra dentro de la etiqueta de inicio** de un elemento e **indican las propiedades** que pueden llevar asociadas los elementos.

Ejemplo en HTML:

- Hola

Ejemplo en XML:

- <ccaa capital="Sevilla" poblacion="8,427M">Andalucia</ccaa>
- <ccaa poblacion="8,427M">Andalucia
- <capital>Sevilla</capital>
- </ccaa>

Atributos

Permiten añadir propiedades a los elementos de un documento. Los atributos:

- **No pueden organizarse** en ninguna **jerarquía**.
- **No** pueden **contener ningún otro elemento o atributo** y no reflejan ninguna estructura lógica.
- **No** se debe **utilizar un atributo para contener información susceptible de ser dividido**.

Los **atributos** se definen y dan valor dentro de una etiqueta de inicio o de elemento vacío, a continuación del nombre del elemento o de la definición de otro atributo siempre separado de ellos por un espacio.

Los valores del atributo van precedidos de un igual que sigue al nombre del mismo y tienen que definirse entre comillas simples o dobles.

Los nombres de los atributos han de cumplir las mismas reglas que los de los elementos, y no pueden contener el carácter menor que, <.

El prólogo

Si se incluye, **el prólogo debe preceder al ejemplar del documento**. Su inclusión **facilita el procesado de la información** del ejemplar. El prólogo está dividido en dos partes:

- **La declaración XML**: En el caso de incluirse ha de ser la primera línea del documento, de no ser así se genera un error que impide que el documento sea procesado. El hecho de que sea opcional permite el procesamiento de documentos HTML y SGML como si fueran XML, si fuera obligatoria éstos deberían incluir una declaración de versión XML que no tienen. El prólogo puede tener tres funciones:
  - **Declaración la versión de XML** usada para elaborar el documento. Para ello se utiliza la etiqueta: *<?xml version= "1.0" ?>* En este caso indica que el documento fue creado para la versión 1.0 de XML.
  - **Declaración de la codificación empleada para representar los caracteres**. Determina el conjunto de caracteres que se utiliza en el documento. Para ello se escribe: <?xml version= "1.0" encoding="UTF-8" ?> En este caso se usa el código UTF-8
  - **Declaración de la autonomía del documento**. Informa de si el documento necesita de otro para su interpretación. Para declararlo hay que definir el prólogo completo: <?xml version= "1.0" encoding="iso-8859-1" standalone="yes" ?> En este caso, el documento es independiente, de no ser así el atributo standalone hubiese tomado el valor "no". Este atributo no tiene valor por defecto, por lo que si no se especifica, el documento XML no puede validarse.
- **La declaración del tipo de documento**, define qué tipo de documento estamos creando para ser procesado correctamente. Toda declaración de tipo de documento comienza por la cadena: <!DOCTYPE Nombre_tipo ...>

Es la parte más importante de un documento XML, ya que contiene los datos reales del documento. Está formado por elementos anidados.

Los elementos son los distintos bloques de información que permiten definir la estructura de un documento XML. Está, delimitados por una **etiqueta** de apertura y una etiqueta de cierre. A su vez los elementos pueden estar formados por otros elementos y/o por **atributos**.

En realidad, el ejemplar es el elemento raíz de un documento XML. Todos los datos de un documento XML han de pertenecer a un elemento del mismo.

Los nombres de las etiquetas han de ser autodescriptivos, lo que facilita el trabajo que se hace con ellas.

El ejemplar: Algunas Reglas

La formación de elementos ha de cumplir ciertas normas para que queden perfectamente definidos y que el documento XML al que pertenecen pueda ser interpretado por los procesadores XML sin generar ningún error fatal. Dichas reglas son:

- En todo documento XML debe existir **un elemento raíz, y sólo uno**.
- Todos los elementos **tienen una etiqueta de inicio y otra de cierre**. En el caso de que en el documento existan elementos vacíos, se pueden sustituir las etiquetas de inicio y cierre por una de elemento vacío. Ésta se construye como la etiqueta de inicio, pero sustituyendo el carácter ">" por "/>. Es decir, <elemento></elemento> puede sustituirse por: <elemento/>
- Al anidar elementos hay que tener en cuenta que **no puede cerrarse un elemento que contenga algún otro elemento que aún no se haya cerrado**.
- **Los nombres de las etiquetas de inicio y de cierre de un mismo elemento han de ser idénticos.** Pueden ser cualquier cadena alfanumérica que no contenga espacios y no comience ni por el carácter dos puntos, ":", ni por la cadena "xml" ni ninguna de sus versiones en que se cambien mayúsculas y minúsculas ("XML", "XmL", "xML",...). Las últimas especificaciones emitidas por el W3C indican la necesidad de que vayan escritas siempre en minúsculas para considerar que el documento está correctamente creado
- El contenido de los elementos no puede contener la cadena "]]>" por compatibilidad con SGML. Además no se pueden utilizar directamente los caracteres mayor que, >, menor que, <, ampersand, &, dobles comillas, ", y apostrofe, ‘. En el caso de tener que utilizar estos caracteres se sustituyen por las siguientes cadenas: > (>) < (<) & (&) “ (") ‘ (')
- Para utilizar caracteres especiales, como £, ©, ®,... hay que usar las expresiones &#D; o &#H; donde D y H se corresponden respectivamente con el número decimal o hexadecimal correspondiente al carácter que se quiere representar en el código UNICODE. Por ejemplo, para incluir el carácter de Euro, €, se usarían las cadenas € o €

Ejemplo

```
<direccion>
    <usuario>
        <titulo>Sra.</titulo>
        <nombre>Jose Manuel</nombre>
        <apellidos>Perez</apellidos>
    </usuario>
    <calle>Ermita  15</calle>
    <ciudad provincia="Madrid">Leganes</ciudad>
    <cp>28932</cp>
</direccion>
```

Elemento nombre

Elemento calle

Elemento ciudad

Elemento cp

En el ejemplo anterior, el elemento <nombre > tiene tres elementos hijos: <titulo>,<nombre> y <apellidos> y estado es un atributo del elemento <ciudad>

Herramientas de edición

Los lenguajes de marcas son ficheros de texto plano, por lo que se puede usar cualquier herramienta de edición de texto para su creación/modificación. Por ejemplo el bloc de notas, el Notepad++ o Atom nos permite editar lenguajes de marca.

Documentos XML bien formados

Documentos XML bien formados

Todos los documentos XML deben verificar las reglas sintácticas que define la recomendación del W3C para el estándar XML. **Esas normas básicas son**:

- El documento ha de tener **definido** un **prólogo** con la declaración xml completa.
- Existe **un único elemento raíz** para cada documento: es un solo elemento en el que todos los demás elementos y contenidos se encuentran anidados.
- Hay que **cumplir las reglas sintácticas** del lenguaje XML para definir los distintos elementos y atributos del documento

Espacios de nombres en XML

https://docs.microsoft.com/es-es/dotnet/standard/data/xml/managing-namespaces-in-an-xml-document

Espacios de nombre en XML

Permiten definir la pertenencia de los elementos y los atributos de un documento XML al contexto de un vocabulario XML. De **este modo se resuelven las ambigüedades** que se pueden producir al juntar dos documentos distintos, de dos autores diferentes, que han utilizado el mismo nombre de etiqueta para representar cosas distintas.

Los espacios de nombres también conocidos como ***namespaces***, permiten dar un nombre único a cada elemento, indexándolos según el nombre del vocabulario adecuado además están asociados a un URI que los identifica de forma única.

En el documento, las etiquetas ambiguas se sustituyen por otras en las que el nombre del elemento está precedido de un prefijo, que determina el contexto al que pertenece la etiqueta, seguido de dos puntos. Esto es: <prefijo:nombre_etiqueta></prefijo:nombre_etiqueta>

Esta etiqueta se denomina "nombre cualificado". Al definir el prefijo hay que tener en cuenta que no se pueden utilizar espacios ni caracteres especiales y que no puede comenzar por un digito.

Antes de poder utilizar un prefijo de un espacio de nombres, para resolver la ambigüedad de dos o más etiquetas, es necesario declarar el espacio de nombres, es decir, asociar un índice con el URI asignado al espacio de nombres, mediante un atributo especial xmlns. Esto se hace entre el prólogo y el ejemplar de un documento XML y su sintaxis es la siguiente: <conexion>://<direccionservidor>/<apartado1>/<apartado2>/...

Ejemplo

Sean los documentos XML que organizan la información sobre los profesores y los alumnos de un curso respectivamente:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes“ ?>
<!DOCTYPE alumnos>
<alumnos>
   <nombre>José Manuel Pérez</nombre>
   <nombre>Isabel González Fernández</nombre>
   <nombre>Ricardo Martínez López</nombre>
</alumnos>
```

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<!DOCTYPE profesores>
<profesores>
   <nombre>Pilar Ruiz Pérez</nombre>
   <nombre>Tomás Rodríguez Hernández</nombre>
</profesores>
```

Al hacer un documento sobre los miembros del curso no se distinguirían los profesores de los alumnos, para resolverlo definiremos un espacio de nombres para cada contexto:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<!DOCTYPE miembros>
<alumnos xmlns:alumnos=”http://curso/alumnos”>
<profesores xmlns:profesores=”http://curso/profesores”>
<asistentes>
   <alumnos:nombre>José Manuel Pérez</alumnos:nombre>
   <alumnos:nombre>Isabel González Fernández</alumnos:nombre>
   <alumnos:nombre>Ricardo Martínez López</alumnos:nombre>
   <profesores:nombre>Pilar Ruiz Pérez</profesores:nombre>
   <profesores:nombre>Tomás Rodríguez Hernández</profesores:nombre>
</asistentes>
```