

UD3.1 Definición de esquemas y vocabularios

Contenido

1. Introducción
2. ¿Qué es XML?
3. Principales usos de XML
4. Conceptos básicos de XML
 - 4.1. Recordatorio de HTML
 - 4.2. Creación de un primer documento XML
 - 4.3. Las ventajas de XML
5. La estructura de XML
6. La sintaxis de XML
7. Documentos bien formados



1. Introducción

- A finales de los años 80 del pasado siglo el lenguaje de marcas por excelencia era **SGML**, un lenguaje completo que permitía definir la estructura de la información así como el formato de presentación que tendría.
- Sin embargo, a pesar de sus múltiples virtudes, **SGML** presentaba un importante inconveniente, era demasiado complejo para integrarse en la Web. Sin embargo, respondía a la vez a las restricciones de presentación y de descripción de datos.

```
<!DOCTYPE motd [ <!E
<motd>
<!-- created: 2003-12-12-->
<sentence>Do not throw
out the <keep>baby</>
with the
<refuse>dirty</>,
<refuse>stinky</>,
<refuse>bathwater</>.
</>
<!-- finish this later-->
</motd>
```

1. Introducción

- Simplificado y adaptado, primero dio a luz al **HTML**, aprovechando su capacidad para mostrar información.
- A finales de 1996 la W3C se encontraba frente a un importante reto, la evolución de la web y otras tecnologías asociadas habían puesto de manifiesto la necesidad de un estándar para la descripción y transmisión de información. Así, se intentó llegar a un compromiso entre la sencillez que ofrecía HTML y la capacidad semántica de SGML.
- El SGML 'light' que crearon fue bautizado primero como **MGML** (por Minimal Generalized Markup Language), **SLIM** (por Structured Language for Internet Markup), MAGMA (por Minimal Architecture for Generalized Markup Applications) y luego finalmente **XML** (por eXtensible Markup Language). En febrero de 1998, XML 1.0 se convierte en recomendación oficial del W3C.

1. Introducción

- La gran preocupación de **SGML** era la de **separar el contenido de un documento de su formato**.
- Los **documentos** así creados estaban **destinados a imprimirse**. Gracias a este concepto, un documento (que contiene los datos) se podría imprimir en varios formatos.
- **SGML era tan complejo** que las manipulaciones sólo podían llevarse a cabo por especialistas (en general, redactores técnicos).
- Poco a poco **se democratiza la documentación online**, introduciendo al mismo tiempo el principio del **hipervínculo**.

1. Introducción

- Desde 1992, gracias a la llegada de la Web y del lenguaje HTML (HyperText Markup Language), los documentos electrónicos de hipertexto se han publicado de manera extensa. Por otra parte, la disponibilidad de motores de búsqueda ha demostrado la eficacia de la aplicación de tales documentos.
- Por consiguiente, el SGML era demasiado complejo y por lo tanto no apto para la Web, y el HTML no era muy escalable. **El W3C naturalmente quería diseñar un lenguaje que poseyera la facilidad de implementación del HTML y la riqueza semántica del SGML. Así nació el XML.**

2. ¿Qué es XML?

- **XML** (eXtended Markup Language) es un metalenguaje extensible de etiquetas desarrollado por la W3C que permite definir la gramática de lenguajes específicos (De la misma manera que HTML es un lenguaje definido a partir de SGML) .
- Por tanto, XML no es un lenguaje en particular, sino una manera de **definir lenguajes para diferentes necesidades**.
- De este modo, XML se puede ver como un subconjunto (mucho más simple) de SGML.
- La especificación de XML es una décima parte de la de SGML

3. Principales usos de XML

- **XML como base de datos.** → El uso principal de XML es estructurar datos, es por ello que en los últimos años han aparecido multitud de lenguajes de bases de datos basados en él.
- **Sindicación de noticias**
- **XML para el almacenamiento de gráficos vectoriales.**
- **Fórmulas matemáticas con XML** → MathML
- **Estructuras moleculares e información científica y química.** → CML
- **XML para intercambio de información** → Servicios Web
- **XML para la definición de interfaces gráficas de usuario**

4. Conceptos básicos de XML

4.1. Recordatorio sobre HTML

- **A diferencia del HTML, el XML no está limitado por un etiquetado predefinido.**
- Dentro de un documento XML, **tenemos total libertad para definir nuestras propias etiquetas.**
- Recordemos que un documento HTML es un documento de texto compuesto de parejas de "etiqueta de apertura/etiqueta de cierre" que describen una función en la organización del documento.
- Los documentos HTML se pueden editar con un simple editor de texto y se interpretan por los navegadores web.

4. Conceptos básicos de XML

4.1. Recordatorio sobre HTML

- Las siguientes líneas describen un documento HTML muy simple que presenta un texto en negrita en la ventana del navegador (la pantalla siguiente presenta la ejecución del documento en un navegador web).

```
<HTML>
<HEAD>
  <TITLE>La página HTML más sencilla</TITLE>
</HEAD>
<BODY>
  <B> Una página HTML sencilla </B>
</BODY>
</HTML>
```



4. Conceptos básicos de XML

4.2. Creación de un primer documento XML

- En un documento XML puede crear sus propias etiquetas de descripción.

```
<nombreapellido>Alberto Oliva</nombreapellido>
```

- Esta línea es sintácticamente correcta en XML, pero no se puede utilizar en un documento HTML.
- Los elementos creados pueden contener nuevos elementos.

```
<identidad>  
  <apellido>Oliva</apellido>  
  <nombre>Alberto</nombre>  
</identidad>
```

4. Conceptos básicos de XML

4.2. Creación de un primer documento XML

- Un documento XML es un archivo de texto con la extensión ".xml" que se puede editar en cualquier editor de texto.
- Se compone de etiquetas personalizables. Sin embargo, a diferencia del HTML, el resultado puede reutilizarse.

4. Conceptos básicos de XML

4.2. Creación de un primer documento XML

- Las siguientes líneas de código presentan un documento XML que describe una tarjeta de visita:

```
<?xml version="1.0"?>
<tarjetadevisita>
  <apellido>LAVALLE</apellido>
  <nombre>Christophe</nombre>
  <empresa/>
  <puesto>Doctor</puesto>
  <direccion>
    <numero>19</numero>
    <calle>Avda de Florida</calle>
    <codigopostal>28008</codigopostal>
    <ciudad>MADRID</ciudad>
  </direccion>
  <numdetelefono>914357265</numdetelefono>
  <numdemovil/>
  <numdefax/>
  <email/>
</tarjetadevisita>
```

4. Conceptos básicos de XML

4.2. Creación de un primer documento XML

Ejemplo 1.

Copia el código de la anterior diapositiva en tu editor de texto y guárdalo como ejemplo1.xml

Ábrelo con tu navegador de internet favorito. ¿Qué aparece?

Modifica ahora el documento eliminando la etiqueta de cierre `</numdetelefono>`

¿Qué ha sucedido?

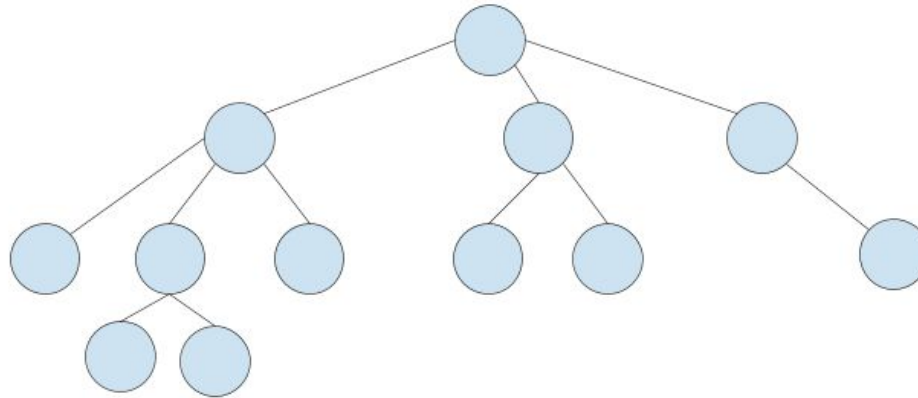
4. Conceptos básicos de XML

4.3. Las ventajas de XML

- XML es una alternativa eficaz a HTML para la adaptación a nuevos entornos más complejos (comercio electrónico, mensajería, etc.). XML busca, principalmente, disociar los datos de su presentación.
- El lenguaje XML incluye varios archivos, cada uno con una función específica.
 - **El archivo XML** (eXtensible Markup Language) contiene el conjunto de datos (archivos .xml).
 - **El DTD** (Document Type Definition) contiene la definición de las características de los elementos que se van a mostrar (archivos .dtd).
 - **Los archivos CSS**
 - **Los archivos XSL** (XML Stylesheet Language) son hojas de estilo mucho más potentes que las CSS y mejor adaptadas al XML (archivos .xsl).

5. La estructura jerárquica de XML

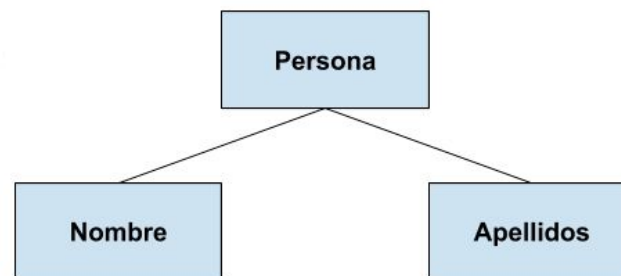
- En un documento XML la información se organiza de manera jerárquica, de manera que los elementos se relacionan entre sí mediante relaciones de padres, hijos, hermanos, ascendientes, descendientes, etc.



5. La estructura jerárquica de XML

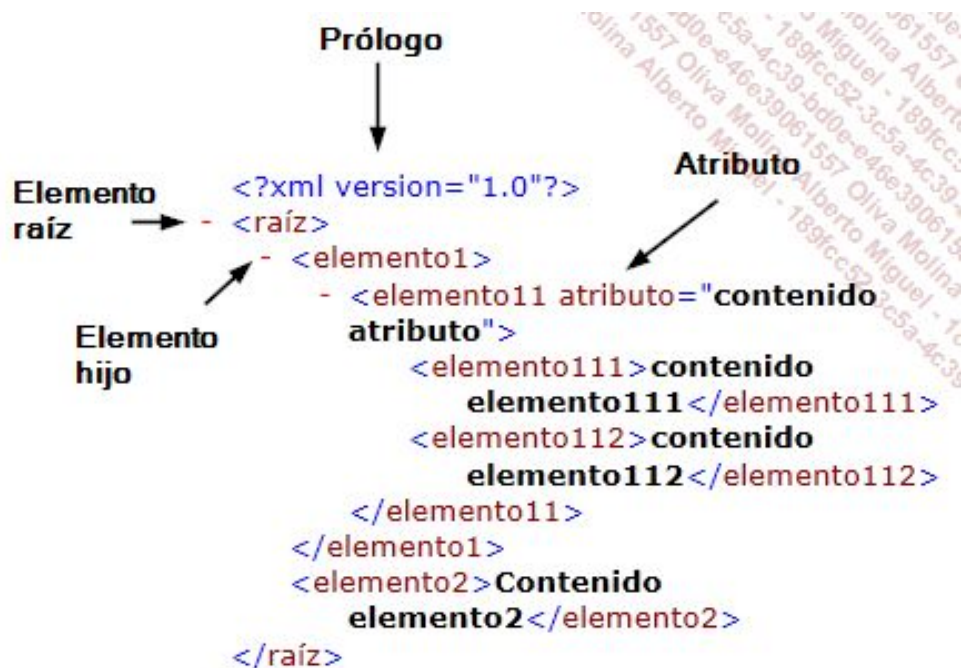
- Ejemplo: El elemento `<persona>` es padre de `<nombre>` y `<apellidos>` por lo que podríamos decir que `<nombre>` y `<apellidos>` son hermanos.
- De mismo modo podemos decir que `<nombre>` y `<apellidos>` son descendientes de `<persona>` y que `<persona>` es ascendiente de `<nombre>` y `<apellidos>`

```
<persona>  
  <nombre>Alberto</nombre>  
  <apellidos>Oliva</apellidos>  
</persona>
```



6. La sintaxis XML

- El archivo XML está organizado según varios componentes que pueden representarse en la forma de un árbol.
- El siguiente esquema enumera todos estos componentes.



6. La sintaxis XML

La primera línea del documento XML

- La primera línea de un documento XML es el **prólogo** y contiene, entre otras cosas, la **declaración de la versión de XML** utilizada para la descripción de la información. Esta declaración tiene el siguiente formato:

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

- También se incluye la codificación de caracteres utilizados** en el documento.
- La escritura de los caracteres XML cumple **con la norma UNICODE**, desarrollada para facilitar la internacionalización de los archivos.
- El prólogo contiene **también a la información DTD para la validación y las hojas de estilo CSS o XSL para la representación de datos**. Estudiaremos estos conceptos más adelante.

6. La sintaxis XML

Los elementos

- Los elementos forman la estructura en árbol de un documento XML. El nombre de los elementos se incluye dentro de las etiquetas, su valor se coloca entre una etiqueta de apertura y una etiqueta de cierre:

`<NombreDeElemento>Valor del elemento</NombreDeElemento>`

- Un elemento vacío se refiere a un elemento cuyo valor no se ha rellenado. Existen dos formas de especificar un elemento de este tipo:

`<elementovacío></elementovacío>`

`<elementovacío/>`

6. La sintaxis XML

Los elementos

- Los nombres de los elementos están sujetos a varias reglas. A menudo son causantes de errores:
 - Sensibilidad a mayúsculas y minúsculas → XML es case sensitive, lo que significa que **debemos respetar las mayúsculas y minúsculas del nombre**. Por tanto, la siguiente línea es incorrecta:

```
<Elemento></ELEMENTO>
```

- La correcta sería:

```
<Elemento></Elemento>
```

- Por convención, los nombres de los elementos deben aparecer siempre en minúsculas.

6. La sintaxis XML

Los elementos

- Los nombres de los elementos están sujetos a varias reglas. A menudo son causantes de errores (II):
 - **No se permiten espacios en los nombres de los elementos**, por tanto, debemos **usar** notación **camelCase** o **guiones bajos** en caso de que contengan más de una palabra.

```
<nombreDelElemento></nombreDelElemento>  
<nombre_del_elemento></nombre_del_elemento>
```

- A pesar de que UTF-8 permite el uso de caracteres especiales como vocales acentuadas o ñ, ¿, etc. se desaconseja totalmente su uso.

6. La sintaxis XML

Los atributos

- Los atributos son información adicional relacionada con los elementos XML.
- La escritura que se utiliza para especificar un atributo de un elemento es la siguiente:

```
<elemento atributo="valor del atributo">valor</elemento>
```

- El objetivo de los atributos es proporcionar información adicional sobre un elemento concreto.
- **Se pueden usar tanto atributos como elementos para representar información. Sin embargo un número excesivo de atributos puede provocar que el documento sea menos legible, más difícil de mantener y difícilmente extensible.**
- **Por tanto, relegamos el uso de atributos a la representación de información secundaria o metadatos.**

6. La sintaxis XML

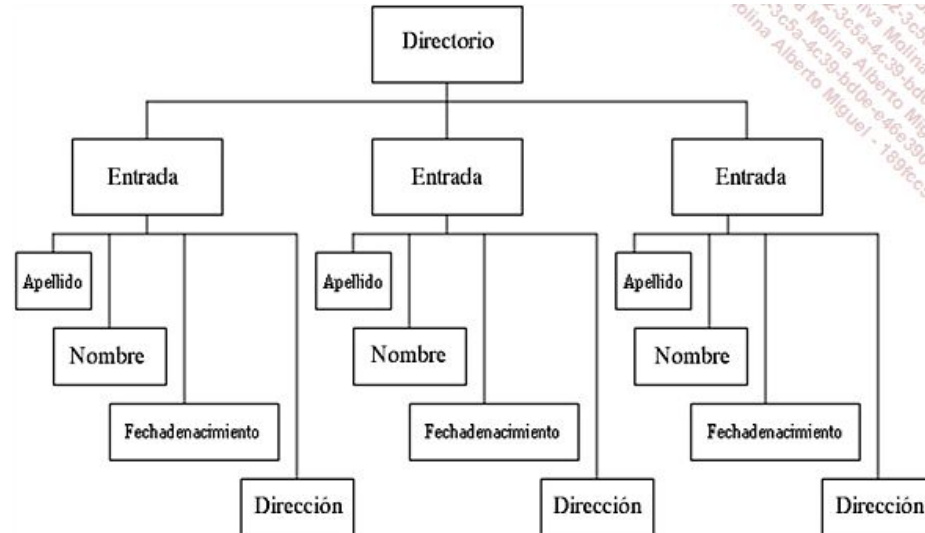
Uso de atributos frente a elementos

- Elementos:
 - Se emplean para representar contenido o jerarquías de contenido.
 - **Se pueden extender con otros elementos en su interior.**
 - **El orden en que aparecen es representativo.**
 - Pueden tener atributos.
 - Puede haber múltiples ocurrencias de un elemento.
- Atributos
 - Van asociados a elementos.
 - Son modificadores de la información
 - Su uso principal es la representación de metadatos
 - **Su orden no es representativo.**

6. La sintaxis XML

La raíz

- Todo documento XML debe contener una raíz: la raíz es el elemento padre de todos los demás elementos.
- El siguiente esquema permite entender el concepto de raíz del documento. Representa el árbol de elementos que permiten construir el siguiente documento XML.



6. La sintaxis XML

La raíz

- El siguiente documento XML implementa el esquema anterior. El elemento padre es `<directorio>`.
- Da lugar a elementos de tipo `<entrada>`.

```
<?xml version="1.0" encoding="UTF-8"?>
<directorio>
  <entrada>
    <apellido>BUCKLEY</apellido>
    <nombre>Jeff</nombre>
    <fechadenacimiento />
    <dirección />
  </entrada>

  <entrada>
    <apellido>Mitchell</apellido>
    <nombre>Mitch</nombre>
    <fechadenacimiento />
    <dirección />
  </entrada>

  <entrada>
    <apellido>HENDRIX</apellido>
    <nombre>Jimi</nombre>
    <fechadenacimiento />
    <dirección />
  </entrada>
</directorio>
```

6. La sintaxis XML

Los comentarios

- Como en todos los lenguajes, los desarrolladores de XML pueden añadir comentarios de diseño.
- Estos comentarios no los procesa el compilador, pero para ello deben incluirse en el documento XML de la siguiente manera:

<!-- Escriba aquí su comentario -->

6. La sintaxis XML

Caracteres especiales

- En XML algunos símbolos están reservados para su uso en casos específicos por lo, para poder representarlos es necesario conocer su codificación.

Código	Carácter
&	&
<	<
>	>
'	'
&aquot;	"

6. La sintaxis XML

Secciones CDATA

- Son conjuntos de caracteres que el procesador no debe analizar (Similares a los comentarios).
- La definición de estas secciones permite agilizar el análisis del documento y deja libertad del autor para introducir caracteres como < y &.

```
<?xml version="1.0" encoding="UTF-8"?>
<ejemplos>
  <ejemplo>
    <nombre>Ejemplo1.html</nombre>
    <codigo><![CDATA[
      <html>
        <body>
          <h3>Titulo</h3>
        </body>
      </html>
    ]]>
    </codigo>
  </ejemplo>
</ejemplos>
```

6. La sintaxis XML

El atributo standalone

- En el prólogo de un documento XML, además de los atributos **version** y **encoding**, también se puede escribir el atributo **standalone**, que puede tomar dos valores ("yes" o "no").
- Al escribir **standalone="yes"** se está indicando que el documento es independiente de otros, como por ejemplo de una DTD (Document Type Definition, Definición de Tipo de Documento) externa. En caso contrario, significará que el documento no es independiente.

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

6. La sintaxis XML

Texto

- El texto, como tipo de nodo que representa datos en XML puede aparecer como contenido de un elemento o como valor de un atributo. En ningún otro sitio.
- Los espacios en blanco reciben un tratamiento especial:
 - Tabulador → \t
 - Nueva línea → \n
 - Retorno de carro → \r
 - Espacio simple → \s

6. La sintaxis XML

Espacios de nombres

- Es un mecanismo para evitar conflictos cuando mezclamos diferentes vocabularios (Lenguajes) dentro de un mismo documento XML.
- Los trataremos en profundidad más adelante.

7. Documentos bien formados.

- **La especificación XML define la sintaxis que el lenguaje debe seguir:**
 - Como se delimitan los elementos con etiquetas.
 - Qué formato puede tener una etiqueta.
 - Que nombres son aceptables para los elementos.
 - Dónde se colocan los atributos.
- **Un documento XML se dice que está bien formado si cumple las reglas establecidas por el W3C en las especificaciones para XML.** Estas reglas son muy numerosas y, en ocasiones, complejas de entender. Vamos a repasar las más significativas.

7. Documentos bien formados.

- El documento puede empezar por una instrucción de procesamiento XML, que indica la versión de XML y, finalmente, la codificación de caracteres y si está listo para procesarse independientemente o requiere de otros archivos externos para esta tarea.
- Debe existir un único elemento raíz. Éste elemento tendrá como descendientes a todos los demás elementos. El documento XML bien formado más simple consta de sólo un elemento raíz, sin ningún descendiente.
- Los elementos que no sean vacíos deben tener una etiqueta apertura y otra de cierre.
- Los elementos vacíos deben cerrarse con `/>`.

7. Documentos bien formados.

- Los nombres de elementos y atributos son sensibles a mayúsculas y minúsculas.
- Los valores de los atributos deben estar comprendidos entre comillas simples o dobles, pero del mismo tipo.
- No puede haber dos atributos con el mismo nombre asociados al mismo elemento.
- No se pueden introducir comentarios en ningún lugar del interior de las etiquetas de apertura y cierre de los elementos.
- No puede haber nada antes del prólogo del documento.
- No pueden aparecer los signos < ni & el contenido textual de elementos ni atributos.

7. Documentos bien formados.

- Los elementos deben aparecer correctamente anidados en cuanto a su apertura y cierre. Es decir, los elementos se cierran en orden inverso al que se abre.

8. Espacios de nombres

- Como sabemos, XML es un metalenguaje de programación, esto permite que, a partir de él, se han definido diferentes lenguajes y vocabularios para usos específicos.
- Sin embargo, sea cual sea el vocabulario que empleemos para construir nuestro documento (En ocasiones más de uno) el formato de fondo siempre será XML.
- ¿Cómo podremos distinguir indicar a que lenguaje o vocabulario pertenecen cada una de las etiquetas de nuestro documento?
 - *A través de los llamados espacios de nombres o namespaces*

8. Espacios de nombres

- Los espacios de nombres (namespaces) permiten definir la pertenencia de los elementos y los atributos de un documento XML a una familia de vocabulario XML.
- Un nombre de elemento puede pertenecer a varias familias sin que por ello tenga que significar lo mismo.
- El espacio de nombres no es esencial en todos los documentos, pero resulta útil cuando se usan etiquetas procedentes de diferentes orígenes (por ejemplo, etiquetas nuevas dentro de un documento XML), o etiquetas que se quieren procesar de forma diferente.
- En general, los espacios de nombres permiten definir cada nombre de elemento de manera única, indexándolos por el nombre de la familia de vocabulario adecuada.

8. Espacios de nombres

- Los dos documentos XML tienen en común el elemento direccion.
- Por el momento son dos vocabularios diferentes pero puede ser que en algún momento tengamos que integrarlos en un mismo documento XML.
- Supongamos que ambos documentos juntos nos ofrecen información sobre la localización de una empresa: su dirección física y su dirección en Internet.
- El elemento dirección podría resultar ambiguo.
- Podríamos renombrarlo y en el segundo documento XML llamarlo ip-servidor, pero no es una buena solución. **Se recomienda utilizar "Namespaces".**

```
<?xml version="1.0" encoding="utf-8"?>
<direccion>
  <calle>Avd. Diagonal 8</calle>
  <ciudad>Barcelona</ciudad>
  <provincia>Barcelona</ciudad>
  <pais>España</pais>
</direccion>
```

```
<?xml version="1.0" encoding="utf-8"?>
<servidor>
  <url>http://www.palotes.com</url>
  <direccion>123.45.67.8</direccion>
</servidor>
```

8. Espacios de nombres

- Antes de poder usar un namespace en nuestro documento debemos *importarlo*.
- Para realizar la importación o carga de un espacio de nombres (namespace) en nuestro documento XML, dentro de cualquier elemento (Óptimamente el elemento raíz) hacemos referencia a él con la siguiente sintaxis:

`<nombre_elemento xmlns:ALIAS_EDN="URI">`

- **xmlns** → Palabra reservada que indica que vamos a importar un espacio de nombres en nuestro documento XML.
- **ALIAS_EDN** → Cuando importamos un namespace debemos indicar alias va a tomar dentro de nuestro documento XML. Este valor es totalmente subjetivo y lo define el programador.
- **URI** → La URI es la dirección del documento de descripción del espacio de nombres.

8. Espacios de nombres

- Los espacios de nombres están adscritos a un identificador **URI** (*Universal Resource Identifier*).
- **Los URI son hipervínculos de acceso a un recurso remoto.** La sintaxis de un URI es la siguiente: <tipodeconexión>://<direccióndelservidor>/<carpeta>/etc.
- Por ejemplo: <https://www.w3.org/TR/html401/>

8. Espacios de nombres

- De este modo, gracias a los espacios de nombres, **es posible integrar etiquetas de diversos lenguajes en un mismo documento** XML (HTML, XHTML, SMIL u otros lenguajes de etiquetas).
- En un mismo documento, para utilizar elementos que tengan el mismo nombre y que pertenezcan a dos lenguajes diferentes, es imprescindible utilizar el concepto de índice (o prefijo) anteponiendo al nombre del elemento o atributo el alias del espacio de nombres seguido del carácter ":" (dos puntos).

8. Espacios de nombres

- En el siguiente ejemplo podemos ver como, gracias al uso de los namespaces, hemos integrado en un mismo documento XML etiquetas pertenecientes a HTML y MathML.

```
<?xml version="1.0"?>
<ejemplo xmlns:H="http://www.w3.org/REC-html40"
          xmlns:M="http://www.w3.org/REC-MathML">
  <H:b>
    <M:mn>2</M:mn>
  </H:b>
</ejemplo>
```

8. Espacios de nombres

- Supongamos, para el siguiente ejemplo, que se han definido dos espacios de nombres, el primero perteneciente a la definición de los datos de un cliente y el segundo a los datos propios de un pedido.

```
<?xml version="1.0" encoding="UTF-8"?>
<cliente xmlns:cli="http://miespacio.org/primer"
  xmlns:ped="http://miespacio.org/segundo" >
  <cli:numero_ID>1232654</cli:numero_ID>
  <cli:nombre>Fulanito de Tal</cli:nombre>
  <cli:telefono>99999999</cli:telefono>
  <ped:pedido>
    <ped:numero_ID>6523213</ped:numero_ID>
    <ped:articulo>Martillo</ped:articulo>
    <ped:precio>187,90</ped:precio>
  </ped:pedido>
</cliente>
```

8. Espacios de nombres

- Un documento XML puede tener tantos espacios de nombres como se quieran declarar, y se pueden mezclar elementos de diferentes espacios de nombres, e incluso sin ningún espacio, tal como se hace a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>
<mc:micasa xmlns:mc='http://www.geneura.org/micasa'
  xmlns:mueble='http://www.geneura.org/mueble'>
  <habitacion id="comedor">
    <mc:mueble>aparador</mc:mueble>
    <mc:mueble>
      <mueble:nombre>Sofá</mueble:nombre>
      <mueble:descripcion>Peludo</mueble:descripcion>
      <mueble:tamano>Inconmensurable</mueble:tamano>
    </mc:mueble>
  </habitacion>
</mc:micasa>
```

8. Espacios de nombres

- Los atributos no están unidos, como opción predeterminada, a ningún espacio de nombres. Pero pueden asociarse utilizando el prefijo del "namespace" que se desea.
 - ❑ **Elemento A** está asociado al xmlns predeterminado ("http://www.thefactory.com/panda")
 - ❑ **Atributo B** está asociado al xmlns http://www.thefactory.com/history"
 - ❑ **Atributo C** no está asociado a ningún espacio de nombres

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<x xmlns="http://www.thefactory.com/panda"
  xmlns:history="http://www.thefactory.com/history">
  <A history:B="foo" C="bar"/>
</x>
```

8. Espacios de nombres

- Unicidad de los atributos
 - El nombre expandido de los atributos está formado por su URI + el propio atributo.
 - **Ningún elemento XML puede tener dos atributos con el mismo nombre expandido.** Incluso si estos atributos pertenecen a diferentes alias.
 - El siguiente caso violaría esta restricción.

```
<?xml version="1.0"?>
<raiz xmlns:ns1="http://ejemplo.org/ns"
xmlns:ns2="http://ejemplo.org/ns">
    <elemento ns1:atributo="hola" ns2:atributo="adios" />
</raiz>
```

8. Espacios de nombres

Espacio de nombres por defecto

- Definir un espacio de nombres por defecto para un elemento nos ahorra tener que volver a definirlo en cada uno de sus hijos.

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```


8. Espacios de nombres

- Conviene usar los espacios de nombres cuando no hay otro remedio, o cuando hay que combinar conjuntos de etiquetas XML procedentes de diferentes procedencias.
- Más adelante, cuando se vea XML Schema o XSLT, los espacios de nombres servirán para especificar qué diccionario de datos usar en cada momento.
- En la diapositiva siguiente se puede ver el uso en un caso real de un documento XML en el que se ha insertado lenguaje XSLT para transformar el documento XML en un HTML y así poder mostrar datos de este documento en un navegador.

8. Espacios de nombres

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

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
<html>
<body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr>
      <th style="text-align:left">Title</th>
      <th style="text-align:left">Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="artist"/></td>
```

Recursos y fuentes

Fuentes:

Tutorial de W3school <https://www.w3schools.com/xml/default.asp>

Tutoriales Abrirllave <https://www.abrirllave.com/>

¿Qué es XML? <https://www.mclibre.org/consultar/xml/lecciones/xml-quees.html>

