UT 3.2 - SINTAXIS Y ESTRUCTURA DE UN DOCUMENTO XML

Índice de contenido

1 INTRODUCCIÓN

Un documento XML está formado con “texto plano”, es decir todos sus caracteres están representados visualmente exceptuando salto de línea, tabulador y espacio.

XML organiza y permite nuevas formas de incluir información extra a través de **marcas** a nuestros textos.

Esta información adicional permite realizar sobre estos textos enriquecidos un gran número de tareas informáticas tales como: búsquedas más precisas, personalización de la información, generación automática de informes, filtrados, etc.

Las recomendaciones de XML oficiales puedes encontrarlas en:

- https://www.w3.org/TR/xml/
- https://www.w3.org/TR/2006/REC-xml11-20060816/

2 MARCAS Y DATOS

La primera clasificación básica que se puede hacer en un documento XML es la distinción entre marcas y datos.

Un documento XML se compone exclusivamente de marcas y datos entremezclados.

Los **datos** son la verdadera información del documento XML.

Las **marcas** añaden información sobre el contenido del documento, sobre su estructura y en general sobre el propio documento, es esa información extra que convierte cualquier texto normal en un documento XML y que hace "visible" a los ordenadores la información de los textos.

Las etiquetas delimitan el texto que desea marcar encerrándolo entre una etiqueta de inicio de marcado < > y una etiqueta de cierre de marcado </ >

```
Ejemplo:    <nombre>Miguel Hernández</nombre>
              <profesion>poeta</profesion>
```

A la construcción **<etiqueta> dato </etiqueta>** se le denomina **elemento** y constituye la base principal de los documentos XML.

Los componentes de un documento XML son:

- Elementos
- Comentarios
- Instrucciones de procesamiento
- Referencias a Entidades
- Secciones de Datos (CData)

3 ESTRUCTURA DE UN DOCUMENTO XML

Un documento XML puede estar formado por un **Prólogo** (es una parte opcional) y un **Ejemplar.**

3.1 EL PRÓLOGO

Es información sobre la información contenida en el documento.

El prólogo es opcional, pero su inclusión es muy recomendable ya que facilita un procesado fiable y robusto de la información contenida en el ejemplar.

El prólogo se compone de una declaración XML y de una declaración de tipo de documento.

3.1.1 LA DECLARACIÓN XML. <?xml ?>

La declaración XML cumple varias funciones:

- Marca el documento como texto XML.
- Declara cuál es la versión de XML utilizada para elaborar el documento a través de lo que se denomina una declaración de versión.
- Aporta información sobre la codificación empleada para representar los caracteres mediante una declaración de codificación.
- ...

Un ejemplo de declaración XML completa podría ser el siguiente:

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

- La **versión** permite indicar la versión para la que se elaboró el documento y permitir que los documentos se adapten a la evolución del estándar.
- La **codificación** nos permite indicar el juego de caracteres utilizado en el documento. El valor por defecto es **UTF-8** .

3.1.2 LA DECLARACIÓN DEL TIPO DE DOCUMENTO. <! DOCTYPE >

La Declaración de Tipo del Documento DTD (Document Type Definition) provee una serie de mecanismos que aportan funcionalidad a XML, gracias a ella es posible definir una serie restricciones y características que deben cumplir los documentos. Se verá en profundidad más adelante.

3.2 EL EJEMPLAR

Es la parte más importante ya que contiene la información del documento, es decir, los datos a los que se les ha añadido el marcado. Es posible definir un tipo para los documentos XML.

Un documento que cumple esos requisitos es una ocurrencia de este tipo, es decir un ejemplar del mismo.

4 ELEMENTOS Y ATRIBUTOS

Son los componentes básicos XML para su principal función: “añadir información sobre la información”. Con ellos se describen estructuras y contextos (o especificación de contenidos) que hacen posible que la información contenida en un documento sea tratada en procesamientos informáticos complejos.

4.1 ELEMENTOS

XML nos permite crear nuestras propias marcas. Se pueden anidar unas con otras y entremezclare con datos carácter para **determinar la estructura del documento dando significado a los diferentes componentes de un texto** (es decir, aportan un contexto). **Estas marcas reciben el nombre de elementos**.

Los elementos permiten introducir información sobre la información para que pueda ser procesada por un ordenador.

Los elementos son los ladrillos, las piezas básicas del rompecabezas.

Los elementos son *"la unidad lógica básica con capacidad para representar la estructura lógica y la semántica del contenido de un documento XML"*.

Los elementos cumplen las siguientes funcionalidades:

- Especificación de contextos
- Delimitación de contenidos
- Estructuración de contenidos
- Jerarquización de elementos

Ejemplo un poema XML:

```
<?xml version="1.0" ?>
<Poema>
    Pablo Neruda
    “Puedo escribir los versos más tristes esta noche”
    Puedo escribir los versos más tristes esta noche.
    Escribir, por ejemplo: "La noche está estrellada,
    y tiritan, azules, los astros, a lo lejos."
    El viento de la noche gira en el cielo y canta.
    Puedo escribir los versos más tristes esta noche.
    Yo la quise, y a veces ella también me quiso.
    ...
```

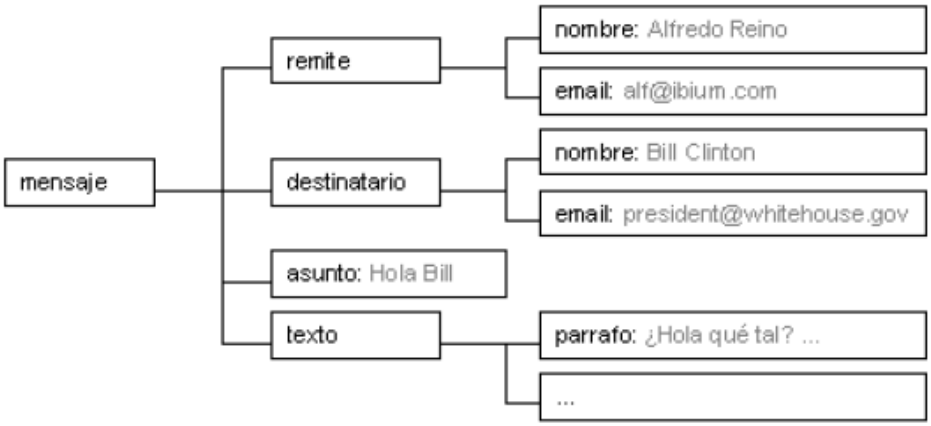
```
</Poema>
```

Éste es un documento XML muy sencillo, está reducido al mínimo y tiene un único elemento, *Poema*, gracias al cual podríamos saber que el documento contiene un poema, y podríamos diferenciarlo de otros documentos como cartas, pedidos, recetas, etc. Lo cierto es que la información que añade este elemento no es demasiada y seguramente no queramos conformarnos con eso. El documento de partida se puede refinar para añadirle otros elementos.

```
<?xml version="1.0" ?>
<poema>
        <autor>Pablo Neruda</autor>
        <titulo>“Puedo escribir los versos más tristes esta noche”</titulo>
        <cuerpo>
                Puedo escribir los versos más tristes esta noche.
                Escribir, por ejemplo: "La noche está estrellada,
                y tiritan, azules, los astros, a lo lejos."
                El viento de la noche gira en el cielo y canta.
                Puedo escribir los versos más tristes esta noche.
                Yo la quise, y a veces ella también me quiso.
                ...
        </cuerpo>
</poema>
```

En este segundo ejemplo existen elementos que identifican el título, autor y el cuerpo del poema, con su ayuda, seríamos capaces, por ejemplo, de listar todos los títulos de los poemas de los que disponemos, o buscar un título determinado y mostrar el cuerpo del poema. Las piezas de texto que forman el título, el autor y el cuerpo están perfectamente identificadas y delimitadas. Este proceso de refinamiento y adición de elementos podría continuarse mientras fuera necesario.

Ejemplo correo electrónico (*fuente: DesarrolloWeb*)




```
<?xml version="1.0" ?>
<mensaje>
        <remite>
                <nombre>Alfredo Reino</nombre>
                <email>alf@ibium.com</email>
        </remite>
        <destinatario>
                <nombre>Bill Clinton</nombre>
                <email>president@whitehouse.gov</email>
        </destinatario>
        <asunto>Hola Bill</asunto>
        <texto>
                <parrafo>
                        ¿Hola qué tal? Hace <enfasis>mucho</enfasis> que no
                        escribes. A ver si llamas y quedamos para tomar algo.
                </parrafo>
        </texto>
</mensaje>
```

4.1.1 LOS TRES COMPONENTES DE UN ELEMENTO

Los elementos están formados por tres componentes: una etiqueta de inicio o apertura(por ejemplo <titulo>) una etiqueta de finalización o cierre (por ejemplo

</titulo>) y un contenido situado entre ambas etiquetas. También existen elementos compuestos por una sólo etiqueta, aquellos que no tienen contenido.

El hecho de olvidar una etiqueta de finalización o de inicio implica un **error de buena formación**. Una buena costumbre para no olvidar una etiqueta de fin es escribir la de inicio y la de fin al mismo tiempo.

4.1.2 EL ELEMENTO RAÍZ

Es un elemento especial que agrupa en su interior al resto de los elementos y datos carácter. Recoge todo el contenido del ejemplar del documento, separándolo de otras partes (si las hubiera). Ejemplos de elemento raíz son <poema> y <mensaje>

Todos los documentos XML tienen obligatoriamente un elemento raíz (uno y sólo uno), un documento XML **sin** el elemento **raíz o con dos o más** elementos raíz **no está bien formado**.

4.1.3 REGLAS DE BUENA FORMACIÓN DE LOS ELEMENTOS

Los elementos al igual que todos los componentes de un documento XML deben seguir ciertas reglas de buena formación, aplicables tanto a las etiquetas de inicio y fin como al contenido que admiten.

Un procesador XML conforme con la *Especificación XML*, comprobará que el documento cumple estas restricciones de buena formación. Cualquier restricción no cumplida será detectada y se tratará como un error fatal. El procesador informará a la aplicación y dejará de trabajar de una manera normal. Fundamentalmente, el objetivo de estas restricciones es asegurar que los documentos XML puedan ser interpretados por los procesadores XML sin ninguna ambigüedad, de manera que todos los elementos (y atributos) quedan perfectamente definidos.

A continuación se presentan algunas de las condiciones que deben cumplir los elementos para considerarse bien formados referentes a la formación de los nombres, a las etiquetas de inicio y cierre y a los caracteres prohibidos.

Formación de los nombres
• Las reglas de formación de los nombres (reglas [4] y [5] de la Recomendación) definen la forma general que debe tener un nombre y básicamente, estas reglas de producción dicen que: [4] NameChar ::= Letter | Digit | '.' | '-' | '_' | ':' | CombiningChar | Extender [5] Name ::= (Letter | '_' | ':') (NameChar)* Un nombre (Name) se compone como mínimo de una letra, (desde la "a" a la "z" o desde la "A" a la "Z"), de un guión bajo o de un carácter de dos puntos.

El carácter inicial (letra, guión bajo o carácter de dos puntos) puede estar seguido de otros caracteres, definidos dentro del grupo NameChar (regla [4]), es decir, una o más letras, dígitos, puntos, guiones, guiones bajos, caracteres de dos puntos, caracteres CombiningChar y caracteres Extender, en cualquier combinación.

El resto de los caracteres y los espacios en blanco no pueden formar parte de un nombre. Observad que un nombre no puede comenzar ni por un dígito, ni por un punto, ni por un guión.

- La codificación de los nombres es sensible a mayúsculas y minúsculas
- Los nombres no pueden empezar por la cadena xml, incluidas todas las combinaciones de minúsculas y mayúsculas, (XML, Xml, xMl, ...), estos nombres se reservan para posteriores estandarizaciones.
- Dice la *Recomendación XML* que el carácter de dos puntos está reservado para la experimentación con espacios de nombres. Esto implica que no debe utilizarse en nombres, salvo con espacios de nombres.
- También especifica la Recomendación que los procesadores de XML deben aceptar este carácter como un carácter de nombre, a pesar de ello, algunos no lo hacen con lo que el carácter de dos puntos debe retirarse del nombre si se desea utilizar ese procesador en concreto.
- Para más información sobre los caracteres incluidos dentro del grupo "CombiningChar" y "Extender", consultar la Recomendación

Resumen:

- Los nombres de elementos pueden tener mayúsculas y minúsculas
- Los nombres de elementos deben comenzar con una letra o un guión bajo
- Los nombres de elementos no pueden comenzar con xml (o XML, o XML, etc.)
- Los nombres de elemento puede contener letras, dígitos, guiones, subrayados y puntos
- Los nombres de elementos no pueden contener espacios

Etiquetas de inicio, de fin y de elemento vacío

- **Etiqueta de inicio**: las etiquetas de inicio comienzan con el carácter <, y a continuación, sin ninguna separación, un identificador genérico (un nombre XML). A lo anterior, le pueden seguir espacios en blanco completamente opcionales. Las etiquetas finalizan con un carácter >. En la etiqueta de inicio se

pueden especificar atributos del modo atributo=“valor”. Todas la etiquetas de inicio deben tener su correspondiente etiqueta de fin con el mismo identificador genérico.

- **Etiqueta de fin**: las etiquetas de fin comienzan con la cadena **</**, seguida, sin separación, de un identificador genérico (un nombre XML) seguido a su vez de un carácter **>**. A continuación del nombre del elemento puede opcionalmente haber espacios en blanco. Para que una etiqueta de fin tenga sentido debe haber una etiqueta de inicio anterior, con el mismo identificador genérico (incluidas las mayúsculas y las minúsculas, *case sensitive*).
- Los elementos sin contenido reciben el nombre de elementos vacíos, estos elementos pueden aparecer expresados de dos maneras, mediante dos etiquetas, una de inicio y otra de fin, seguidas, o mediante una única etiqueta especial, la etiqueta de elemento vacío.
- **Etiqueta de elemento vacío**: está formada por el carácter **<**, seguido de un identificador genérico (un nombre XML) seguido de espacios en blanco opcionales, y finalizando con **/>.** Los elementos vacíos normalmente poseen atributos.

<elemento atributo=”valor” />

Caracteres prohibidos dentro del contenido de los elementos

- Dentro del contenido de los elementos no pueden aparecer los siguientes caracteres y cadenas ya que tienen funciones especiales. Si aparecen deben escaparse para que no se interpreten erróneamente:
  - **<** porque podría confundirse con el comienzo de una etiqueta.
  - **&** porque podría confundirse como comienzo de una referencia a entidad
  - **[]]>]** esto es así para asegurar cierta compatibilidad hacia atrás con SGML.
- La mejor manera para escapar caracteres es utilizar referencias a entidades predefinidas.

4.2 ATRIBUTOS

Los atributos asociados a los elementos actúan como modificadores, como adjetivos que incluyen información adicional sobre un elemento. (En HTML los atributos se utilizan, por ejemplo para poner un borde a una imagen, poner un color de fondo, o alienar texto. Los atributos en XML funcionan de un modo similar.)

Los atributos y sus valores se especifican, dentro de una etiqueta de inicio o de elemento vacío.

Los nombres de los atributos deben cumplir las mismas restricciones que los nombres de los elementos.

Los atributos se expresan en pares *atributo=”valor”*.

A ambos lados del símbolo igual pueden aparecer tantos espacios en blanco como se desee o ninguno.

El valor de un atributo se tiene que escribir entre comillas simples o dobles. El tipo de comillas que no se utilice se puede incluir dentro del valor para marcar literales.

Si un elemento tiene varios atributos éstos se deben separar entre sí por espacios en blanco.

Dentro de la etiqueta de inicio de un elemento no pueden aparecer dos referencias al mismo atributo.

4.2.1 Uso de los atributos

A veces a la hora de diseñar, puede ser complicado elegir entre un atributo o un elemento, si bien muchas veces el resultado es el mismo.

No hay reglas acerca de cuándo utilizar atributos o cuando el uso de elementos en XML

Pero hay que tener en cuenta que los atributos no pueden contener subelementos, ni subatributos, ni se organizan en ninguna jerarquía, por lo que tienen una capacidad de representación mucho más reducida que los elementos y no pueden reflejar una estructura lógica. La información que añaden suele ser de **poca entidad, sencilla, sin estructura y no subdivisible**.

Ejemplo de distintos modelados de un elemento:

```
<perro> <nombre>Rocky</nombre> <raza>pequines</raza></perro>
<perro raza=”pequines”> <nombre> Rocky</nombre> </perro>
<perro raza=”pequines” nombre=”Rocky” />
```

Ejemplo siguiendo recomendación:

```
<alumno>
        <nombre>Jesús</nombre>
        <apellidos>Sánchez Camacho</apellidos>
        <fechaNacimiento>
                <dia>02</dia> <mes>06</mes> <año>1964</año>
        </fechaNacimiento>
```

```
        <sexo>varón</sexo>
        <telefono tipo=”movil”>646721212</telefono>
        <domicilio>
                <direccion>Paseo Puerta del Angel 23</direccion>
                <ciudad>Madrid</ciudad>
                <codigoPostal>28011</codigoPostal>
        </domicilio>
</alumno>
```

5 OTROS COMPONENTES DE UN DOCUMENTO XML

5.1 COMENTARIOS

Los comentarios en los documentos XML empiezan por **<!--** y acaban por **-->**.

Pueden contener cualquier cadena de texto excepto el literal **--**.

Pueden colocarse en cualquier parte del documento.

Ej:

```
<!-- Esto es comentario y puede poner el carácter < ó > ó cualquier otro menos el prohibido -->
```

5.2 SECCIONES DE DATOS CDATA

Le indican al *parser* (analizador sintáctico) que ignore todos los caracteres de marcas que se encuentren en el interior de estas secciones y los trate como datos.

<![CDATA[sección de datos **]]>**

Un documento XML puede contener secciones **CDATA** (Character DATA) para escribir texto que no se desea que sea analizado. Por ejemplo, esto puede ser útil cuando se quiere escribir texto que contenga alguno de los caracteres problemáticos: menor que "<" o ampersand "&".

El único literal que no puede ser utilizado dentro de la sección es **]]>**

No pueden anidarse secciones CDATA.

El PARSER XML: es un pequeño programa/librería/módulo que se encarga de toda la validación y análisis de XML de forma transparente al usuario. Normalmente está

incluido en los navegadores de hoy en día en librerías separadas. El parser lo necesita aquel que tenga que extraer información del documento, es decir el navegador.

EJEMPLO Una sección CDATA puede contener, por ejemplo, el código fuente de un programa escrito en lenguaje C:

```
<?xml version="1.0" encoding="UTF-8"?>
<ejemplo_CDATA>
<![CDATA[
#include <stdio.h>
int main()
{
  float nota;
  printf( "\n   Introduzca nota (real): " );
  scanf( "%f", &nota );
  if ( 5 <= nota )
    printf( "\n   APROBADO" );
  return 0;
}
]]>
</ejemplo_CDATA>
```

Otro ejemplo:

```
<![CDATA[ <!ENTITY amp "&"> <!-- &= ampersand -->
<CODIGO>
*p=&q->campo;
a=(x<y)?33:44;
</CODIGO>
]]>
```

5.3 ENTIDADES

Las entidades (*entity*) se usan en XML básicamente como:

- representación alternativa de los caracteres especiales (como por ejemplo las comillas dobles ó la marca de apertura en un elemento)

- hacer referencia a trozos de texto que se repiten o varían frecuentemente
- incluir el contenido de otros documentos almacenados en ficheros externos

5.3.1 Referencias a entidades

En XML existen algunos caracteres que son especiales por su significado y, para escribirlos en un documento XML, se pueden utilizar las referencias a entidades mostradas en la siguiente tabla:

Referencias a entidades en XML

Carácter	Entidad	Referencia a entidad
< (*menor que*)	lt (*less than*)	<
> (*mayor que*)	gt (*greater than*)	>
" (*comilla doble*)	quot (*quotation mark*)	"
' (*comilla simple*)	apos (*apostrophe*)	'
& (*ampersand*)	amp (*ampersand*)	&

EJEMPLO Dado el archivo ***"entidades.xml"***:

```
<?xml version="1.0" encoding="UTF-8"?>
<entidades>
  <menor_que>&lt;</menor_que>
  <mayor_que>&gt;</mayor_que>
  <comilla_doble>&quot;</comilla_doble>
  <comilla_simple>&apos;</comilla_simple>
  <ampersand>&amp;</ampersand>
</entidades>
```

Referencia a carácter

Una **referencia a carácter** se refiere a un carácter específico en el conjunto de caracteres en el ISO/IEC 10646, por ejemplo uno no directamente accesible desde un dispositivo de entrada.

Su sintaxis es: **&**referencia**;**

La referencia puede ser un nºdecimal (precedido &#) o un nºhexadecimal (precedido &#x)

Ejemplo: & para el símbolo ampersand especificado en decimal.

En un documento XML se pueden escribir referencias de caracteres Unicode con los símbolos &#, seguidos del valor decimal o hexadecimal del carácter Unicode que se quiera representar y, finalmente, añadiendo el carácter *punto y coma* ";".

Representación del carácter Euro (€) en XML

EJEMPLO Dado el documento XML ***"productos.xml"***:

```
<?xml version="1.0" encoding="UTF-8"?>
<productos>
  <nombre precio="12.56&#8364;">Gorro de lana</nombre>
  <nombre precio="16.99&#x20AC;">Gorro polar</nombre>
</productos>
```

5.3.2 Entidades internas

También llamadas macros ó constantes de texto, las entidades internas son las q se asocian a una cadena de caracteres.

Ejemplo1:

```
<!ENTITY nom "Juan Pérez López">
```

Así, si escribo en mi documento &nom; es como si estuviera escribiendo Juan Pérez López.

Ejemplo2:

```
<!ENTITY llavecerrada "&#x125;">  Hará referencia al carácter }
```

Para referenciarlo: &llavecerrada;

5.3.3 Entidades externas

Sirven para referenciar su contenido ó valor desde otros ficheros, los cuales pueden ser binarios (ej,, una imagen jpg) o de texto(ej. fichero xml). Si el contenido del fichero que hace la referencia es de texto, entonces se sustituye tal y como si de una entidad interna se tratara.

Ej.

```
<!ENTITY arch System "arch.txt">
```

Inserta el contenido del fichero arch.txt en lugar en la q se hace la referencia.

Para más información revisar la referencia del lenguaje XML

5.4 INSTRUCCIONES DE PROCESAMIENTO

Son un modo de pasar información a la aplicación que ha de tratar fichero XML. (El parser no la trata) La más común es para indicar la vesión de XML y el juego de caracteres que utiliza el documento. Su formato es **<?nombre datos ?>**

Ejemplo:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

En un documento XML, **una instrucción de procesamiento (*processing instruction*) sirve para indicar cierta información al programa que procese dicho documento.** Las instrucciones de proceso se escriben empezando con la pareja de caracteres "<?" y finalizando con "?>".

5.4.1 Cómo asociar un archivo CSS a un documento XML

EJEMPLO En un documento XML podría escribirse, por ejemplo, la siguiente instrucción de procesamiento:

```
<?xml-stylesheet type="text/css" href="estilo-animales.css"?>
```

Esta instrucción sirve para asociar el archivo **CSS** (*Cascading Style Sheets*, Hojas de Estilo en Cascada) ***"estilo-animales.css"*** al documento XML. Dicho archivo podría contener, por ejemplo, el siguiente código:

```
nombre{color:blue;font-size:40px}
patas{color:red;font-size:22px}
```

De forma que, dado por ejemplo el archivo ***"animales.xml"***:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="estilo-animales.css"?>
<animales>
  <animal>
    <nombre>perro</nombre>
    <patas>4</patas>
  </animal>
  <animal>
    <nombre>pato</nombre>
```

```
    <patas>2</patas>
  </animal>
  <animal>
    <nombre>ballena</nombre>
    <patas>0</patas>
   </animal>
</animales>
```

6 ESPACIOS DE NOMBRES

Generalmente los documentos XML se suelen combinar con otros documentos XML existentes, es decir que se usen módulos XML desarrollados por terceras personas. Esta modularidad es muy importante ya que permite la reutilización de código existente que está ampliamente probado y depurado.

En estos casos es muy fácil que haya elementos con el mismo nombre en diferentes archivos y aparezcan conflictos al unificarlos.

Para evitar este conflicto se utilizan los espacios de nombres que establecen nombres extendidos para los elementos que indique de que archivo original procede.

EJEMPLO Dos documentos XML podrían contener un elemento llamado "carta", pero con significados distintos.

```
<carta>
  <palo>Corazones</palo>
  <numero>7</numero>
</carta>
```

```
<carta>
  <carnes>
    <filete_de_ternera precio="12.95"/>
    <solomillo_a_la_pimienta precio="13.60"/>
  </carnes>
  <pescados>
    <lenguado_al_horno precio="16.20"/>
```

```
    <merluza_en_salsa_verde precio="15.85"/>
  </pescados>
</carta>
```

6.1 Uso del espacio de nombres

De forma que, si se incluyen ambos elementos <carta> en un documento XML, se origina un conflicto de nombres. Para resolverlo, se pueden utilizar espacios de nombres (*XML Namespaces*). Por ejemplo, escribiendo:

```
<?xml version="1.0" encoding="UTF-8"?>
<e1:ejemplo xmlns:e1="http://www.abrirllave.com/ejemplo1"
  xmlns:e2="http://www.abrirllave.com/ejemplo2">
  <e1:carta>
    <e1:palo>Corazones</e1:palo>
    <e1:numero>7</e1:numero>
  </e1:carta>
  <e2:carta>
    <e2:carnes>
      <e2:filete_de_ternera precio="12.95"/>
      <e2:solomillo_a_la_pimienta precio="13.60"/>
    </e2:carnes>
    <e2:pescados>
      <e2:lenguado_al_horno precio="16.20"/>
      <e2:merluza_en_salsa_verde precio="15.85"/>
    </e2:pescados>
  </e2:carta>
</e1:ejemplo>
```

Así los nombres de los elementos estarán compuesto de dos partes: el espacio de nombres, dos puntos y el nombre del elemento.

```
<espacionombres:elemento>contenido</espacionombres:elemento>
```

6.2 Sintaxis para definir un espacio de nombres

Un espacio de nombres se define como una URI

Por lo tanto un para referirse a un elemento hay que utilizar la nomenclatura <uri:elemento>

De forma que, si se incluyen ambos elementos <carta> en un documento XML, se origina un conflicto de nombres. Para resolverlo, se pueden utilizar espacios de nombres (*XML Namespaces*). Por ejemplo, escribiendo:

```
<?xml version="1.0" encoding="UTF-8"?>
<e1:ejemplo xmlns:e1="http://www.abrirllave.com/ejemplo1"
  xmlns:e2="http://www.abrirllave.com/ejemplo2">

  <e1:carta>
    <e1:palo>Corazones</e1:palo>
    <e1:numero>7</e1:numero>
  </e1:carta>

  <e2:carta>
    <e2:carnes>
      <e2:filete_de_ternera precio="12.95"/>
      <e2:solomillo_a_la_pimienta precio="13.60"/>
    </e2:carnes>
    <e2:pescados>
      <e2:lenguado_al_horno precio="16.20"/>
      <e2:merluza_en_salsa_verde precio="15.85"/>
    </e2:pescados>
  </e2:carta>

</e1:ejemplo>
```

Si embargo una URI puede ser muy larga. Para evitar errores y dar claridad se puede utilizar un sinónimo corto para la URI.

Para definir un espacio de nombres se utiliza la siguiente sintaxis:

```
xmlns:prefijo="URI"
```

En el ejemplo, obsérvese que, xmlns es un atributo que se ha utilizado en la etiqueta de inicio del elemento <ejemplo> y, en este caso, se han definido dos espacios de nombres que hacen referencia a los siguientes **URI** (*Uniform Resource Identifier*, Identificador Uniforme de Recurso):

- http://www.abrirllave.com/ejemplo1
- http://www.abrirllave.com/ejemplo2

Los prefijos definidos son e1 y e2, respectivamente. Véase que, se han añadido dichos prefijos a las etiquetas que aparecen en el documento: <e1:carta>, <e2:carta>, <e1:palo>, etc.

Los URI especificados en un documento XML no tienen porqué contener nada, su función es ser únicos. No obstante, en un URI se puede mostrar información si se considera oportuno. Véase, por ejemplo:

- http://www.w3.org/1999/xhtml/
- http://www.w3.org/1999/XSL/Transform
- http://www.w3.org/2000/svg

Un sinónimo se declara con el atributo reservado xmlns con la siguiente sintaxis:

```
<elementoraiz xmlns:sinonimo=”uri”>
        <sinonimo:elemento1>contenido1</sinonimo:elemento1>
        <sinonimo:elemento2>contenido2</sinonimo:elemento2>
<!elementoraiz>
```

Nota: Se utiliza un URI con forma de URL porque es de suponer que el dominio nos pertenece y por tanto hace la URI única. Puede ser que esa URL no apunte a ninguna localización real, y si se navega a ella da error.

6.3 Definición de espacios de nombres en elementos distintos al raíz

EJEMPLO En un documento XML, los espacios de nombres pueden definirse en el elemento raíz –como acabamos de ver– o, directamente, en los elementos que los vayan a utilizar. Por ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<e1:ejemplo xmlns:e1="http://www.abrirllave.com/ejemplo1">
```

```
  <e1:carta>
    <e1:palo>Corazones</e1:palo>
    <e1:numero>7</e1:numero>
  </e1:carta>

  <e2:carta xmlns:e2="http://www.abrirllave.com/ejemplo2">
    <e2:carnes>
      <e2:filete_de_ternera precio="12.95"/>
      <e2:solomillo_a_la_pimienta precio="13.60"/>
    </e2:carnes>
    <e2:pescados>
      <e2:lenguado_al_horno precio="16.20"/>
      <e2:merluza_en_salsa_verde precio="15.85"/>
    </e2:pescados>
  </e2:carta>

</e1:ejemplo>
```

En un documento XML es posible definir todos los espacios de nombres que se necesiten, pudiéndose mezclar –si fuese necesario– los elementos de dichos espacios de nombres.

6.4 Definición de un espacio de nombres por defecto

EJEMPLO Por otra parte, se puede definir un espacio de nombres por defecto mediante la siguiente sintaxis:

```
xmlns="URI"
```

De esta forma, tanto el elemento donde se ha definido el espacio de nombres, como todos sus sucesores (hijos, hijos de hijos, etc.), pertenecerán a dicho espacio de nombres. Por ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<ejemplo xmlns="http://www.abrirllave.com/ejemplo1">
```

```
  <carta>
    <palo>Corazones</palo>
    <numero>7</numero>
  </carta>

</ejemplo>
```

EJEMPLO En el siguiente ejemplo, inicialmente se define un espacio de nombres por defecto para el elemento <ejemplo> y los contenidos en él. Ahora bien, posteriormente, se define un segundo espacio de nombres, que por defecto afecta al segundo elemento <carta> que aparece en el documento y a sus sucesores: <carnes>, <pescados>, <filete_de_ternera>...

```
<?xml version="1.0" encoding="UTF-8"?>
<ejemplo xmlns="http://www.abrirllave.com/ejemplo1">

  <carta>
    <palo>Corazones</palo>
    <numero>7</numero>
  </carta>

  <carta xmlns="http://www.abrirllave.com/ejemplo2">
    <carnes>
      <filete_de_ternera precio="12.95"/>
      <solomillo_a_la_pimienta precio="13.60"/>
    </carnes>
    <pescados>
      <lenguado_al_horno precio="16.20"/>
      <merluza_en_salsa_verde precio="15.85"/>
    </pescados>
  </carta>
```

```
</ejemplo>
```

6.5 Cómo indicar que un elemento no pertenece a ningún espacio de nombres

EJEMPLO En un documento XML, para indicar que determinados elementos –o todos– no pertenecen a ningún espacio de nombres, se escribe el atributo xmlns vacío, es decir, xmlns="".

```
<?xml version="1.0" encoding="UTF-8"?>
<ejemplo xmlns="http://www.abrirllave.com/ejemplo1">

  <carta>
    <palo>Corazones</palo>
    <numero>7</numero>
  </carta>

  <carta xmlns="http://www.abrirllave.com/ejemplo2">
    <carnes>
      <filete_de_ternera precio="12.95"/>
      <solomillo_a_la_pimienta precio="13.60"/>
    </carnes>
    <pescados xmlns="">
      <lenguado_al_horno precio="16.20"/>
      <merluza_en_salsa_verde precio="15.85"/>
    </pescados>
  </carta>

</ejemplo>
```

En este caso, el elemento <pescados> y sus hijos, no pertenecen a ningún espacio de nombres.

7 DOCUMENTOS XML BIEN FORMADOS Y VÁLIDOS

Documento Válido es el que cumple con los requisitos especificados en la definición de su estructura (mediante DTD 0 XML Schema)

Documento Bien Formado es el que cumple con las reglas de formación expuestas:

- El documento debe tener un único elemento raíz
- Los identificadores de elementos y atributos han de cumplir las reglas de nombres XML
- Las etiquetas de los elementos han de estar correctamente anidadas
- Todos los elementos han de tener sus etiquetas de inicio y fin, o ser elementos vacíos
- Los valores de atributos han de estar encerrados entre comillas
- Los elementos y atributos son case-sensitive, sensibles a mayúsculas y minúsculas.
- El documento solo contendrá caracteres válidos dependiendo del tipo de codificación del documento.
- Los caracteres “<”, “>” y “&” solo deben aparecen para delimitar las etiquetas de los elementos y para usar caracteres especiales

8 HERRAMIENTAS QUE COMPRUEBAN LA BUENA FORMACIÓN

Como se decía, es conveniente, al finalizar la creación del documento XML, comprobar que está bien formado. Para ello, hay una gran variedad de aplicaciones disponibles. No se incluyen en este apartado herramientas que necesiten conocimientos sobre programación reservándose este aspecto para capítulos posteriores.

Quizás las aplicaciones más sencillas de utilizar sean los navegadores para la Red que utilizamos diariamente. Si el documento se almacenó en un fichero de texto plano, podemos intentar abrirlo con el explorador Firefox o Microsoft Internet Explorer, si está bien formado podremos ver el árbol asociado al documento.

Para comprobar sintaxis podemos utilizar: nuestro analizador XML (sólo IE), o el analizador en línea de RUWF.

En caso de querer utilizar otras aplicaciones, en las siguientes direcciones podemos utilizar analizadores en línea:

- http://www.w3.org/2001/03/webdata/xsv
- http://validator.w3.org/

- http://www.gotdotnet.com/services/xsdvalidator/
- http://tools.decisionsoft.com/schemaValidate.html
- http://www.asahi-net.or.jp/~eb2m-mrt/onLineValidation.html
- http://www.stg.brown.edu/service/xmlvalid/
- http://www.cogsci.ed.ac.uk/~richard/xml-check.html
- http://msdn.microsoft.com/downloads/samples/internet/xml/xml_validator/