

# Estructura de un documento XML

Antes de trabajar con los documentos XML usando Visual C#, es importante entender correctamente la estructura de este tipo de documento. Los siguientes párrafos van a presentar las nociones elementales que se deben conocer antes de lanzarse a la utilización de documentos XML.

## 1. Partes constituyentes de un documento XML

Un documento XML se puede constituir a partir de los siguientes elementos:

### Instrucción de tratamiento

Las intrucciones de tratamiento permiten incorporar en un documento XML información destinada al procesador XML o a otras aplicaciones antes de editar el documento. Se utilizan estas instrucciones para facilitar una instrucción especial a una aplicación que trabaja en el documento.

Se inserta la instrucción de tratamiento en el documento con la siguiente sintaxis:

```
< ?nombreAplicacion instruccion ?>
```

La primera parte es el nombre de la aplicación a la cual se destina esta instrucción. La segunda parte es el texto de la instrucción.

Un documento XML contiene, en general, una instrucción de tratamiento especial para definir la versión de XML que conforma el documento y la codificación de los caracteres utilizada por el documento.

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

### Comentarios

Los comentarios sirven para incluir en el documento la información destinada a los usuarios de éste. Son ignorados por el procesador XML o por las aplicaciones que usan el documento. No se deben incorporar a una etiqueta.

La siguiente sintaxis se debe utilizar para ubicar un comentario en el documento.

```
<!--esto es un comentario-->
```

En el interior del comentario, la utilización de los caracteres -- está prohibida:

```
<!--esto es -- un comentario-->
```

### Caracteres reservados

Algunos caracteres quedan reservados para el lenguaje XML, como por ejemplo el carácter & del siguiente ejemplo:

```
<menu>Queso & Postres </menu>
```

Para poder utilizar estos caracteres en un documento XML, debe sustituirlos por la siguiente sintaxis:

Carácter	Utilizar en lugar de
&	&amp;
<	&lt;

>	&gt;
'	&apos;
"	&quot;

Por lo tanto, la sintaxis correcta es:

```
<menu>queso &amp; postre</menu>
```

Secuencias de caracteres más largas se pueden incorporar usando una sección CDATA. La sintaxis es la siguiente:

```
<![CDATA[{ Select * from postres where precio < 10} and calorías > 500]]>
```



Con esta sintaxis se puede utilizar cualquier carácter sin necesidad de tomar más precauciones.

## Elementos XML

Un elemento XML es un contenedor que acoge datos y otros elementos. Se compone de una etiqueta de principio y de una etiqueta de fin. La sintaxis de un elemento XML es la siguiente:

```
<NombreElemento>contenido</NombreElemento>
```

Los elementos deben respetar algunas reglas relativas a su forma:

- Los nombres de elementos no pueden contener espacios.
- No pueden empezar por un número o un signo de puntuación.
- No pueden empezar con xml (sea cual sea la caja de la letra).
- Deben empezar justo después del signo >, sin espacio.
- Las etiquetas de principio y fin deben usar de forma idéntica las mayúsculas y las minúsculas.
- Un documento XML debe contener al menos un elemento: el elemento raíz.
- Todos los elementos que siguen el elemento raíz deben anidarse en éste.
- Si un elemento no tiene contenido, puede estar constituido únicamente por una etiqueta de fin.
- Sólo el elemento raíz debe tener una etiqueta de principio y una de fin, incluso aunque no tenga contenido.

### Ejemplo:

```
<?xml version="1.0" encoding="utf-8" ?>
<restaurante>
<menu>
  <entradas>
    <nombre>rábanos</nombre>
    <nombre>pasta</nombre>
    <nombre>salchichón</nombre>
  </entradas>
  <platos>
```

```

    <nombre>paella</nombre>
    <nombre>fideuá</nombre>
    <nombre>uscús</nombre>
</platos>
<quesos>
    <nombre>manchego</nombre>
    <nombre>tetilla</nombre>
    <nombre>cabrales</nombre>
</quesos>
<postres>
    <nombre>helado</nombre>
    <nombre>tarta</nombre>
    <nombre>crema catalana</nombre>
</postres>
</menu>
</restaurante>

```

### **Atributos de elementos**

Se utilizan los atributos de elementos para calificar un elemento. Se ubican en la etiqueta de principio del elemento. Como los elementos, deben seguir unas reglas:

- Un atributo se compone de un nombre y de una asignación de valor.
- Un elemento puede contener un número cualquiera de atributos.
- Los nombres de atributos están separados por espacios.
- Un nombre de atributo sólo puede aparecer una vez en un elemento.
- Un nombre de atributo puede aparecer en varios elementos.
- Un nombre de atributo no puede contener espacios.
- La asignación de un valor a un atributo se hace con el signo igual seguido del valor, rodeado de comillas dobles:

#### **Ejemplo:**

```

<?xml version="1.0" encoding="utf-8" ?>
<restaurante>
<menu type="gastronomico">
    <entradas>
        <nombre calorías="50">rábanos</nombre>
        <nombre calorías="300">pasta</nombre>
        <nombre calorías="350">salchichón</nombre>
    </entradas>
    <platos>
        <nombre calorías="1000">paella</nombre>
        <nombre calorías="2000">fideuá</nombre>
        <nombre calorías="1700">uscús</nombre>
    </platos>
    <quesos>
        <nombre calorías="240">manchego</nombre>
        <nombre calorías="300">tetilla</nombre>
    </quesos>
</menu>
</restaurante>

```

```

    <nombre calorias="120">cabrales</nombre>
</quesos>
<postres>
    <nombre calorias="340" sabor="chocolate">helado</nombre>
    <nombre calorias="250" frutas="manzanas">tarta</nombre>
    <nombre calorias="400">crema catalana</nombre>
</postres>
</menu>
</restaurante>

```

## Espacios de nombres

Un espacio de nombres es un conjunto de nombres de elementos identificados por una referencia única. Permiten evitar las confusiones cuando unos datos XML se fusionan a partir de diferentes fuentes.

Tomemos el siguiente ejemplo, que podría ser un archivo de configuración de una aplicación:

```

<?xml version="1.0" encoding="utf-8" ?>
<aplicacion>
    <menu nombre="archivo">
        <entradas>nuevo</entradas>
        <entradas>abrir</entradas>
        <entradas>cerrar</entradas>
    </menu>
    <menu nombre="edicion">
        <entradas>copiar</entradas>
        <entradas>cortar</entradas>
        <entradas>pegar</entradas>
    </menu>
</aplicacion>

```

En este archivo, tenemos elementos que ya habíamos definido en otro archivo. Es obvio que los elementos **menu** y **entradas** no tienen el mismo significado que en el archivo utilizado anteriormente. Para evitar toda ambigüedad, hay que añadir en cada uno de los archivos una definición de espacio de nombres que convierta cada elemento en único. La definición de un espacio de nombres se efectúa con el atributo `xmlns` seguido de un prefijo y del identificador del espacio de nombres.

La sintaxis es la siguiente para cada uno de nuestros dos archivos:

```

<restaurante xmlns:restaurante="http://www.eni-escuela.es/restaurante">
<aplicacion xmlns:appli="http://www.eni-escuela.es/configappli">

```

Es muy importante que los identificadores de espacios de nombres sean únicos si se desea intercambiar información con otras personas. Por eso, es habitual utilizar el nombre de dominio de la empresa en el identificador (se supone que éste es único). Con esta modificación, podemos usar en el mismo archivo elementos **menu** y **entradas**, añadiendo delante el prefijo del espacio nombres en el cual tienen un significado.

```

<fusion xmlns:appli="http://www.eni-escuela.es/configappli" xmlns:restaurante=
"http://www.eni-escuela.es/restaurante">
    <appli:menu nombre="archivo">
        <appli:entradas>registrar</appli:entradas>
    </appli:menu>
    <restaurante:menu nombre="economico">
        <restaurante:entradas>aguacate</restaurante:entradas>
    </restaurante:menu>
</fusion>

```

## 2. Documento bien formado y documento válido

Gracias al lenguaje XML, tenemos la posibilidad de crear fácilmente documentos estructurados y comprensibles. Existen dos nociones que permiten verificar la calidad de un documento XML: un documento puede estar bien formado y un documento puede ser válido.

### a. Documento bien formado

Un documento está bien formado si obedece a las reglas sintácticas del lenguaje XML. Estas reglas son mucho menos estrictas que las reglas de validez. Gestionan las atribuciones de nombres, las creaciones y las relaciones entre elementos. Para poder ser tratado por un procesador XML, un documento debe estar bien formado. Si el procesador detecta un error, detiene inmediatamente el tratamiento del documento.

### b. Documento válido

Un documento válido es un documento XML que tiene vinculada una DTD o un esquema XSD (definición del tipo de documento) y respeta todas las reglas de construcción definidas en esta última. Cuando un procesador XML analiza el documento, busca en la DTD o en el esquema XSD una definición para cualquier elemento, atributo, entidad de este documento. En cuanto encuentra un error, detiene el tratamiento.