# 1. Introducción

**XML** (extensible markup language) es una tecnología que sirve como formato de almacenamiento e intercambio de información.

Se usa para almacenamiento de datos, actualización de software, intercambio de información, sindicación de contenidos, etc.

Se verá su estructura y sintaxis, instrucciones de procesamiento, marcas, elementos y atributos, así como otros aspectos relevantes.

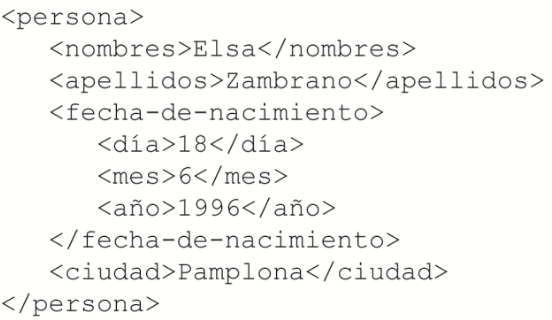
Verificar que están escritos correctamente y son válidos son aspectos fundamentales a la hora de trabajar con estos documentos (comprobar que están bien formado), siguiendo una serie de reglas y normas que pueden encontrarse en la **DTD** (document type definition) o **XSD** (XML schema definition).



**XML** deriva de **SGML** (standard generalized markup language), siendo legible y comprensible para todos, lo que permite que usuarios sin conocimientos de lenguajes de marcas sean capaces de interpretar qué información es la que se almacena en los documentos.

Aunque un documento DTD es más fácil de crear que uno XSD y siguen creándose en la actualidad, presenta una serie de limitaciones: no permite especificar restricciones como rangos concretos, número de caracteres que deben aparecer, expresiones regulares… y otras limitaciones que veremos más adelante.

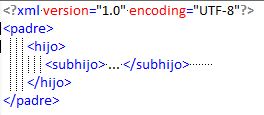
Aspecto de un documento XML:



# 2. Estructura y sintaxis

La estructura de un documento XML es jerárquica arborescente, que contiene a un padre (raíz) único y una serie de hijos (elementos) secundarios. La raíz se sitúa en la parte superior, colgando el resto de elementos de él.

La extensión de este tipo de documentos es .xml.



### Ejercicio 1

Busca por Internet un documento XML y analiza su estructura.

## Ejercicio 2

En el ejemplo siguiente se aprecia un XML que contiene valores sobre especies de árboles, almacenando información como nombre, variedad, origen, etc.

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

<arbol>

<especie>

<nombre>**Litchi**</nombre>

<variedad>**Kway May Pink**</variedad>

<origen>**Filipinas**</origen>

<color\_fruto>**rojo**</color\_fruto>

<maduracion>**agosto**</maduracion>

<riego>**diario**</riego>

<precio>**3.5**</precio>

</especie>

</arbol>

Añade dos especies más al ejemplo.

## 2.1. Marcas o etiquetas, elementos, atributos, comentarios y sección CDATA.

Las **marcas**, también llamadas **etiquetas**, van entre los signos ‘<’ y ‘>’ (como en HTML).

Ejemplo: <nombre>

Habrás podido observar que:

* Las marcas no pueden contener espacio en blanco, comillas simples o dobles, punto y coma, signo de porcentaje, etc.
* Las marcas son sensibles a mayúsculas y minúsculas.

Los **elementos** están compuestos por etiquetas de inicio y fin. En el siguiente ejemplo definimos el elemento nombre.

Ejemplo: <nombre> Litchi </nombre>

Pueden existir **elementos vacíos**.

Ejemplo:

<casado></casado>

<telefono n=”944000333” />

Los atributos se usan para asignar propiedades a los elementos. Se pueden especificar varios a un mismo elemento. No es recomendable usar muchos porque el documento sería menos legible.

Ejemplo:

<nombre dni=”77777777F”>Alumno</nombre>

### Ejercicio 3

Dado el siguiente fichero XML de una agenda personal, transforma los elementos ciudad y email en atributos

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

<agenda>

<contacto nombre=”alumno”>

<ciudad>**Málaga**</ciudad>

<telefono>**666888999**</telefono>

<email>**a@b.com**</email>

</contacto>

</agenda>

Los comentarios sirven de ayuda cuando el documento es extenso o se pretende que todo se entienda con claridad. Pueden introducirse en cualquier parte, excepto dentro de las etiquetas o dentro de otros comentarios o declaraciones.

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

<agenda>

<!--Vamos a crear un contacto-->

<contacto nombre=”alumno”>

<ciudad>**Málaga**</ciudad>

<telefono>**666888999**</telefono>

<email>**a@b.com**</email>

</contacto>

</agenda>

La sección CDATA permite añadir contenido sin ser procesado o analizado. Debe aparecer dentro del documento raiz de un documento XML. Es similar a un comentario. Sintaxis:

<![CDATA[ comentario que no analiza el parser ]]>

### Ejercicio 4

Es preciso mandar una tabla de manera segura y fiable a un programa cliente que recoge información sobre unos árboles subtropicales: variedad, color de fruto, maduración,…

Transforma el siguiente cuadro a XML:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nombre** | **Variedad** | **Ciudad origen** | **Color del fruto** | **Maduración** | **Riego** | **Precio Kg** |
| **Litchi** | Kway may pink | Filipinas | Rosa | Agosto | Diario | 3,5 € |
| **Longan** | Champoo | China | Marrón | Octubre | Diario | 2,5 € |
| **Litchi** | Mauritius | Florida | Rojo | Agosto | Diario | 3 € |

## 3. DTD

Un **DTD** (definición de tipo de documento) es un documento de texto plano donde se especifica o definen una serie de normas que se usan para saber cómo tiene que crearse un fichero XML para poder validarlo.

Aunque no es obligatoria su definición, es recomendable si la información va a ser compartida. De esta manera, los usuarios que trabajen con el XML sabrán las reglas a seguir.

El DTD se crea, preferiblemente, antes que el XML. De esta forma sabremos las normas que hemos de seguir.

Un DTD está formado por elementos, atributos y entidades:

## 3.1. Elementos

Son la base de su estructura. En él se especifica el nodo padre y, entre paréntesis, sus hijos (si los tuviera) separados por una coma.

Ejemplo:

<!ELEMENT nombreElemento (elemento\_hijo1, elemento\_hijo2, …)>

Por otra parte, cada uno de los elementos debe ser de un tipo específico:

<!ELEMENT nombreElemento Tipo>

Siendo tipo:

* **(#PCDATA)**: parsed character data, que indica que el contenido de ese elemento es de tipo texto y es analizado por un parser.
* **(#CDATA)**: character data, que indica que el contenido de ese elemento no es analizado por un parser.
* **ANY**: los elementos pueden contener cualquier valor.
* **EMPTY**: los elementos no tienen contenido. Elemento vacío.

**Ejemplo:**

Se va a crear un documento DTD para el envío de correos electrónicos que contenga los campos asunto, receptor y cuerpo.

|  |  |  |
| --- | --- | --- |
| **Receptor** | **Asunto** | **Cuerpo** |
| [**a@b.com**](mailto:a@b.com) | Cumpleaños | Desearte un feliz… |
| [**c@d.com**](mailto:c@d.com) | Día libre | Estimado Señor X… |

Un fichero DTD para un XML podría ser:

<!ELEMENT bdcorreo (email)+>

<!ELEMENT email (receptor, asunto, cuerpo)>

<!ELEMENT receptor (#PCDATA)>

<!ELEMENT asunto (#PCDATA)>

<!ELEMENT cuerpo (#PCDATA)>

El signo más del ejercicio anterior no es una errata. Se está especificando el **número de veces que puede aparecer el elemento en el XML.**

### Caracteres de ocurrencia de los elementos

|  |  |
| --- | --- |
| **Carácter** | **Descripción** |
| **?** | El elemento puede aparecer cero o una vez (opcional). |
| **\*** | El elemento puede aparecer cero, una o varias veces (opcional). |
| **+** | El elemento debe aparecer una o más veces (obligatorio). |
| **|** | Similar al OR lógico. Significa que puede añadirse uno de los dos elementos que se especifiquen, pero no ambos:  <!ELEMENT nombreE(hijo1, (hijo2|hijo3))> |

Ejemplo:

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

<!DOCTYPE BDcomponentes [

<!ELEMENT BDcomponentes (componente)+>

<!ELEMENT componente (nombre, (precio|precioiva),tipo+,medida)>

<!ELEMENT nombre (#PCDATA)>

<!ELEMENT precio (#PCDATA)>

<!ELEMENT precioiva (#PCDATA)>

<!ELEMENT tipo (#PCDATA)>

<!ELEMENT medida (#PCDATA)>

]>

<BDcomponentes>

<componente>

<nombre>**resistencia**</ nombre >

<precio>**0.25**</ precio >

<tipo>**pasivo**</ tipo >

<tipo>**bobinada**</ tipo >

<medida>**ohmios**</medida>

</ componente >

<componente>

<nombre>**condensador**</ nombre >

<precio>**0.75**</ precio >

<tipo>**pasivo**</ tipo >

<medida>**Faradios**</medida>

</ componente >

</ BDcomponentes >

Si ahora se modifica el ejemplo anterior y no se añade al componente el elemento tipo, no será válido e informará del error.

Para solucionar el problema, podría cambiarse modificando el tipo (en vez con el signo +, con el \*).

Probadlo.

## 3.2. Atributos

Los atributos en una DTD se usan para enriquecer o añadir información a un elemento. Se declaran usando la marca <!ATTLIST Elemento Atributo Tipo Valor.

Ejemplo:

<!ATTLIST usuario clave CDATA #REQUIRED dni CDATA #REQUIRED>

Donde:

* **Elemento**: es el nombre del elemento al que quiere añadirse un atributo.
* **Atributo**: nombre del atributo por añadir.
* **Tipo**: se selecciona un tipo de la tabla 3.2.1.
* **Valor**: comportamiento del valor del atributo (ver tabla 3.2.2).

|  |  |
| --- | --- |
| **Tipo atributo** | **Explicación** |
| **CDATA** | El valor es alfanumérico (texto). |
| **Enumeraciones** | Permite especificar varios valores para un atributo:  <ATTLIST numero octal (0|1|2|3|4|5|6|7) CDATA #REQUIRED> |
| **ENTITY** | El atributo es una entidad que se ha definido previamente. |
| **IDREF** | Hace referencia al identificador de otro elemento. |
| **IDREFS** | Hace referencia a un conjunto de identificadores (ej1 | ej2 | ej3). |
| **ID** | Su contenido es único, identificando de manera inequívoca a cada elemento. |
| **NOTATION** | Se especifican datos que no son XML. |
| **ENTITIES** | Conjunto o lista de entidades. |
| **NMTOKENS** | Permite especificar el valor de un atributo mediante caracteres permitidos. |

Tabla 3.2.1

Para especificar si los valores de los atributos son obligatorios u opcionales, se usan los modificadores de la siguiente tabla:

|  |  |
| --- | --- |
| **Valor** | **Significado** |
| **#FIXED** | Debe especificarse un valor que es fijo y no podrá cambiarse. |
| **#IMPLIED** | El atributo es opcional. |
| **#DEFAULT** | Debe especificarse un valor por defecto que sí podrá cambiarse. |
| **#REQUIRED** | El atributo es obligatorio. |

Tabla 3.2.2

Ejemplo:

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

<!DOCTYPE bdusuarios [

<!ELEMENT bdusuarios (usuario)+>

<!ELEMENT usuario (nombre,email,fechanac)>

<!ATTLIST usuario clave CDATA #REQUIRED dni CDATA #REQUIRED>

<!ELEMENT nombre (#PCDATA)>

<!ELEMENT email (#PCDATA)>

<!ELEMENT fechanac EMPTY>

]>

<bdusuarios>

<usuario clave=”\*\*\*” dni=”\*\*\*\*\*”>

<nombre>**Ataulfo G. Pascual**</ nombre >

<email>**user@e.com**</ email>

<fechanac> </ fechanac>

</usuario>

<usuario clave=”\*\*\*” dni=”\*\*\*\*\*”>

<nombre>**Francisco G. García**</ nombre >

<email>**usr@f.com**</ email>

<fechanac> </ fechanac>

</usuario>

</bdusuarios>

## 3.3. Entidades

Las entidades se consideran constantes y pueden usarse para abreviar texto.

Lo primero es crear una entidad mediante ENTITY, para posteriormente establecer una clave y su valor. Lo vemos claro con un ejemplo:

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

<!DOCTYPE bdusuarios [

<!ELEMENT bdusuarios (usuario)+>

<!ELEMENT usuario (nombre,email,suceso)>

<!ELEMENT nombre (#PCDATA)>

<!ELEMENT email (#PCDATA)>

<!ELEMENT suceso (#PCDATA)>

<!ENTITY web “http://www.google.com”>

<!ENTITY cliente “Se informa que el usuario ha accedido al…”>

]>

<bdusuarios>

<usuario>

<nombre>**Ataulfo G. Pascual**</ nombre >

<email>**user@e.com**</ email>

<suceso>&cliente; registro en la página &web;</suceso>

</usuario>

<usuario>

<nombre>**Francisco G. García**</ nombre >

<email>**usr@f.com**</ email>

<suceso>&cliente; contenido de &web; para consultar el libro X</suceso>

</usuario>

</bdusuarios>

Al visualizarlo en un navegador, la referencia a la entidad del documento XML se transforma en el valor asociado.

### Asociar un DTD a un documento XML

En los ejemplos anteriores se observa que utilizamos la cláusula DOCTYPE para hacer referencia al DTD de un XML. En él se especifica el elemento raíz y entre corchetes los distintos elementos.

Pero también podemos hacer referencia a un DTD externamente, como se ve en el siguiente ejemplo:

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

<!DOCTYPE bdcorreo SYSTEM “midtd.dtd”>

<bdcorreo>

<email>

<receptor> **user@e.com** </receptor>

<asunto>Cumpleaños</asunto>

<cuerpo>Desearte un feliz…</cuerpo>

</email>

<email>

<receptor> **user@f.com** </receptor>

<asunto>Día libre</asunto>

<cuerpo>Estimado Señor X,…</cuerpo>

</email>

</bdcorreo>

El fichero DTD creado se llamará midtd.dtd y contendrá lo siguiente:

<!ELEMENT bdcorreo (email)+>

<!ELEMENT email (receptor,asunto,cuerpo)>

<!ELEMENT receptor (#PCDATA)>

<!ELEMENT asunto (#PCDATA)>

<!ELEMENT cuerpo (#PCDATA)>

### Ejercicio 5

Dado el siguiente DTD. Obtén un XML asociado:

<!ELEMENT arbol (especie)+>

<!ELEMENT especie (nombre,variedad,(origen)+,color\_fruto,(maduracion)+,(riego)?,(precio)?)>

<!ELEMENT nombre (#PCDATA)>

<!ELEMENT variedad (#PCDATA)>

<!ELEMENT origen (#PCDATA)>

<!ELEMENT color\_fruto (#PCDATA)>

<!ELEMENT maduracion (#PCDATA)>

<!ELEMENT riego (#PCDATA)>

<!ELEMENT precio (#PCDATA)>