ORENGA RAMON, ANDREU

1º DAW/DAM

xml/dtd

Contenido

[¿Qué es XML? 3](#_Toc150617845)

[Principales características de XML 3](#_Toc150617846)

[Xml no es 3](#_Toc150617847)

[En XML 4](#_Toc150617848)

[Análisis XML 4](#_Toc150617849)

[Errores en XML 4](#_Toc150617850)

[Analizadores XML 4](#_Toc150617851)

[Editores XML 4](#_Toc150617852)

[XML Reglas Básicas 5](#_Toc150617853)

[Reglas básicas 5](#_Toc150617854)

[Etiqueta raíz: 5](#_Toc150617855)

[Las etiquetas se tienen que cerrar 5](#_Toc150617856)

[Correctamente anidadas 5](#_Toc150617857)

[Nombres correctos 5](#_Toc150617858)

[Case sensitive 5](#_Toc150617859)

[Valores entre comillas (“ ”) 5](#_Toc150617860)

[XML Creación de Documentos 6](#_Toc150617861)

[Declaración 6](#_Toc150617862)

[Etiquetas 6](#_Toc150617863)

[Atributos 6](#_Toc150617864)

[Entidades 7](#_Toc150617865)

[CDATA 7](#_Toc150617866)

[Comentarios 8](#_Toc150617867)

[Instrucciones de proceso 8](#_Toc150617868)

[Como asociar un archivo css a un documento xml 8](#_Toc150617869)

[Espacios de nombres (namespace) 8](#_Toc150617870)

[Esquemas y vocabularios en xml 9](#_Toc150617871)

[DTD 9](#_Toc150617872)

[Sintaxis basica 9](#_Toc150617873)

[DTD Interno 10](#_Toc150617874)

[DTD externo 10](#_Toc150617875)

[Estructura DTD 10](#_Toc150617876)

[Declaracion de elementos 10](#_Toc150617877)

[Tipos de contenidos 11](#_Toc150617878)

[Modelo de contenidos 11](#_Toc150617879)

[Quantificadores 11](#_Toc150617880)

[Atributos 12](#_Toc150617881)

[Declaracion de atributos en dtd 12](#_Toc150617882)

[Tipos de atributos 13](#_Toc150617883)

[Valores de atributos 13](#_Toc150617884)

[Valores predeterminados de atributos 13](#_Toc150617885)

[Entidades 13](#_Toc150617886)

[Uso de las entidades 13](#_Toc150617887)

[Ventajas de las entidades 14](#_Toc150617888)

[Entidades internas 14](#_Toc150617889)

[Entidades generales externas analizadas 14](#_Toc150617890)

[Entidades no analizadas 14](#_Toc150617891)

[Ejemplo DTD 15](#_Toc150617892)

XML/DTD

# ¿Qué es XML?

Es un lkenguaje de marcas utilizado para almacenar y intercanviar datos estructurados de manera legible tanto por humanos como por maquinas.

## Principales características de XML

* Es un lenguaje de marcas: Utiliza etiquetas para definir la estructura de los datos. Elementos que envuelven los datos y describen lo que representan. <persona> (envuelve los datos de una persona.
* Extensible: Que podemos definir nuestras propias etiquetas y estructuras de datos.
* Jerarquizado: Sigue una estructura jerárquica, las etiquetas pueden contener otras etiquetas, creando una estructura en forma de árbol
* Legible para los humanos: Esta en un formato legible para los humanos. Facilita la inspección y la edición manual.
* Interoperabilidad: es un estándar altamente aceptado, comprensible por muchas aplicaciones y plataformas diferentes. Facilita el intercambio de datos entre sistemas diversos.
* Usos variados: se hace servir en una amplia gama de aplicaciones, bases de datos, documentos html, configuraciones de software etc.
* Compatible con Unicode: puede almacenar textos en diversas codificaciones incluyendo Unicode.
* Es versátil y esta estandarizado

## Xml no es

* Un lenguaje de programación.
* No es un protocolo de red.
* No es una base de datos.

Xml no es simplemente un sistema de marcas para estructurar datos, sino que también incluye tecnologías relacionadas que facilitan la creación, el almacenamiento, la transmisión y el procesamiento de datos.

* xPath y XQuery: utilizado para seleccionar y consultar datos dentro de documentos xml.
* XPointer: identifica partes especificas de un documento xml.
* XLink: proporciona una estructura para añadir enlaces de hipertexto.
* DTD: define la estructura de validación, permite asegurar que los datos siguen una estructura especifica
* XSL: se utiliza para transformar documentos xml en otros formatos, como HTML.

## En XML

* Los datos se pueden representar en árboles.
* Todos los documentos tienen una etiqueta padre (raíz) que no es hijo de nadie.
* Las hojas del árbol son los datos contenidos en el documento.
* Están formados por arboles de elementos. Empieza por un elemento raíz, del cual van saliendo las ramas, todos los elementos pueden tener elementos hijos.
* Todos los elementos pueden contener contenido textual y atributos.

## Análisis XML

El objetivo principal de tener reglas es que pueden ser leídas y interpretadas por un ordenador, se utilizan analizadores:

* Comprueban que se cumplen las reglas del xml.
* Comprueban que el documento este “bien formado”.
* Comprueban que el documento es “valido”.

Hay 2 operaciones de comprobación:

1. Correcto: comprueba que el documento sigue las reglas de creación (bien formado).
2. Valido: que el documento cumpla la estructura especifica definida en un esquema o DTD.

## Errores en XML

Errores leves: violación de algunas normas que están marcadas como recomendación (el programa puede intentar continuar.

Errores graves: no se cumplas algunas de las reglas obligatorias: no seguir la sintaxis etc… El programa tendrá que detenerse inmediatamente.

## Analizadores XML

* Xmllint
* Apache Xerces
* Expat
* Waindows.Data.XML

## Editores XML

* Se pueden crear con cualquier editor de texto.
* Servirá de ayuda hacer servir algún editor con soporte xml (autocompletado, validación, creación de esquemas etc.
* Hay editores específicos para trabajar en xml (XML Copy Editor)
* Muchas veces IDEs de programación tienen soporte ( IntelliJ, Eclipse, Webstorm etc.)
* Si tenemos que hacer ediciones complejas podemos hacer servir editores especializados (oXigen XML, Liquid XML Studio, Stylus 6 XML).

# XML Reglas Básicas

Texto, etiquetas y atributos

<nombre cargo = “coordinadora”> Isabel S. </nombre>

* Etiqueta: <nombre></nombre>
* Atributo: cargo=” coordinadora”
* Texto: Isabel S.

No se propone nada sobre cómo se utilizarán los datos, el usuario tiene el control.

## Reglas básicas

### Etiqueta raíz:

* Es la etiqueta que contiene todas las demás .
* Lo único que puede estar fuera de la raíz son los comentarios y las instrucciones de proceso.

### Las etiquetas se tienen que cerrar

* Cada una de las etiquetas se tiene que cerrar ( </etiqueta> )
* Si tenemos en etiquetas sin datos el cierre será />

### Tabla Descripción generada automáticamenteCorrectamente anidadas

### Nombres correctos

* Tienen que empezar por una letra del alfabeto, subrayado, guion o dos puntos.
* No pueden tener espacios.
* No pueden empezar por la palabra xml.

### Case sensitive

Esto indica que xml diferencia entre las letras mayúsculas de las minúsculas.

### Valores entre comillas (“ ”)

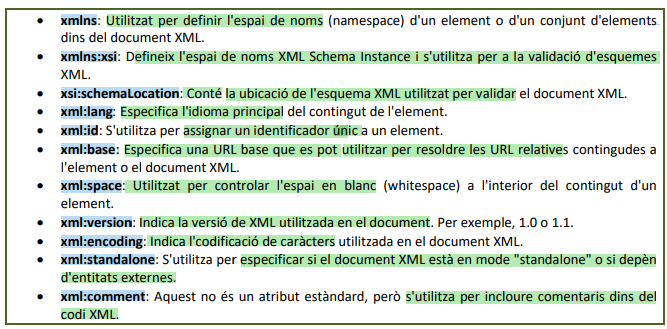
* Los valores de los atributos tienen que estar entre comillas.
* Los atributos siempre tienen que tener algún valor
* No importa si son comillas simples o dobles.

# XML Creación de Documentos

## Declaración

La declaración es una línea especial que se introduce al principio del documento XML para identificar su versión y codificación de caracteres.

La declaración no es obligatoria pero si recomendable. En muchos casos solo contendrá la versión y el encoding.

* <?XML: indica que se trata de un documento XML.
* Versión: generalmente se utiliza la versión 1.0.
* Encoding: define el código de caracteres que hacemos servir, se hace servir Unicode por defecto (UTF-8 y UTF-16) pero se pueden utilizar otros:
* Standalone: indica si el documento XML es independiente de otros recursos externos (yes/no), yes (el documento esta completo) no (el documento no se puede entender por si solo.

## Etiquetas

* Siempre van entre los símbolos “< “ y “>”.
* Son creadas por el autor y tienen que describir los datos que contienen.
* Dentro de una etiqueta podemos tener tantos atributos como necesitemos.

## Atributos

Son elementos de metadatos que se pueden añadir a las etiquetas para proporcionar información adicional o controlar el comportamiento de cómo se procesa.

* Los atributos son parejas formadas por llave=valor.
* Tiene que ser un nombre XML valido y que no esté repetido.
* Podemos crear tantos atributos como necesitemos.
* Pueden estar entre comillas simples o dobles
* Todos los atributos deben tener alguna cosa asignada.

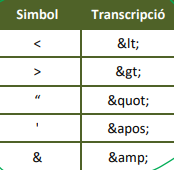
Se utilizan para añadir metadatos o control de procesamiento y presentación del documento XML.

* Metadatos: proporcionan información sobre un elemento.
* Control de contenido: influyen en como se procesa o se presenta el contenido de un elemento (ej. href: es una etiqueta enlace que puede especificar una URL.
* Idioma i codificación: especifica el idioma i la codificación de caracteres de un XML.
* Control de estilos: los atributos como “style” o “class” controlan el estilo CSS.

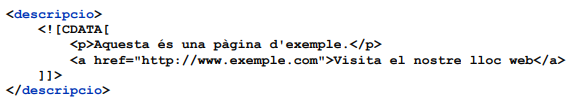
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteAtributos predefinidos:

## Entidades

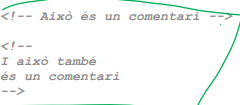
Las entidades en XML son mecanismos que permiten representar y referenciar caracteres especiales. Las mas comunes son:

## CDATA

* CDATA (Carácter de Datos), es una sección especial que permite incluir texto o datos con caracteres especiales.
* Es útil para añadir fragmentos de texto que no quieres que sea analizado o interpretado como parte de la estructura del XML.

El contenido dentro del bloque CDATA se trata como contenido y no como marcas del XML. En el ejemplo todo el contenido HTML se conserva sin ninguna interpretación, puede ser útil cuando queremos almacenar fragmentos de código con caracteres que podrían ser interpretados por el XML.

## Comentarios

Se pueden añadir comentarios que serán ignorados por el programa que procese el documento.

## Instrucciones de proceso

Permiten que programas externos realicen tareas específicas. Se delimita por las marcas <? ¿> i el primer elemento dentro de ellas indica que programa tiene que procesar la instrucción.

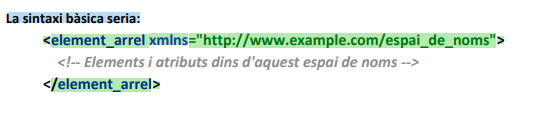
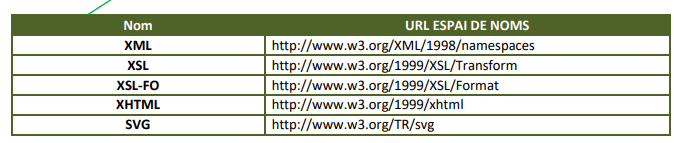
## Como asociar un archivo css a un documento xml

Esta instrucción es un ejemplo de como asociar un archivo CSS a un documento XML para aplicarle estilos.

## Espacios de nombres (namespace)

Cada persona puede crear tantas etiquetas como quiera con el nombre que quiera. Por lo tanto nos podemos encontrar etiquetas repetidas al combinar documentos XML, que pueden significar cosas diferentes y crear conflictos.

Los espacios de nombres aseguran que los nombres de elementos y atributos sean únicos y no entren en conflicto.

Se declaran mediante el uso del atributo xmlns (XML Namespace)

# Esquemas y vocabularios en xml

* Los esquemas XML son como formularios que definen la estructura y el tipo de datos que se han de utilizar en el documento XML .
* El vocabulario XML son un conjunto de palabras y frases que tenemos que utilizar. Definen las etiquetas y elementos específicos que podemos utilizar en nuestro documento XML.

Se utilizan “lenguajes de definición de vocabulario” o “lenguajes de esquemas” (DTD).

* Un documento XML puede:
* Estar bien formado, cumplir la sintaxis.
* Ser valido, cumplir determinadas reglas y normas.
* Para establecer unas reglas de construcción en XML utilizamos DTD.
* Validar es un proceso habitual que realizaremos con programas especiales (procesadores o validadores.

# DTD

Un DTD es una especificación que define la estructura, los elementos y los atributos permitidos en documentos XML, proporcionando una guía para validar.

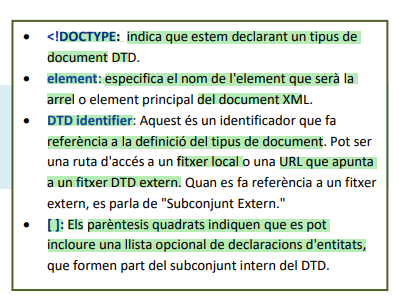
El DTD define:

* Tipo de elementos.
* Atributos.
* Entidades permitidas.
* También se pueden expresar restricciones.

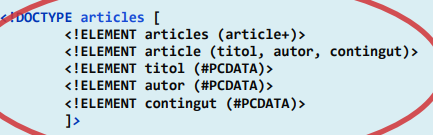
Podemos crear DTDs de dos maneras:

* Ficheros externos: que se pueden compartir con diversos documentos.
* En el propio documento XML (inline)

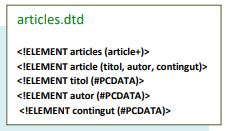
## Sintaxis basica



## DTD Interno

El DTD se define dentro del propio documento XML.

## DTD externo

En este caso el DTD se define en un archivo separado y se asocia al documento XLM (.dtd).

También puede estar en una ubicación externa que puede ser:

* Privada, es la más común, la definimos nosotros y se utiliza con SYSTEM.
* Publica, la define un organismo de estandarización, se utiliza PUBLIC.

## Estructura DTD

* Elementos
* Atributos
* Entidades
* Anotaciones

## Declaracion de elementos



* Nombre elemento: el elemento que se puede definir.
* Tipo de contenido: indica que tipo de contenido puede tener el elemento (PCDATA, EMPTY, ANY etc.).

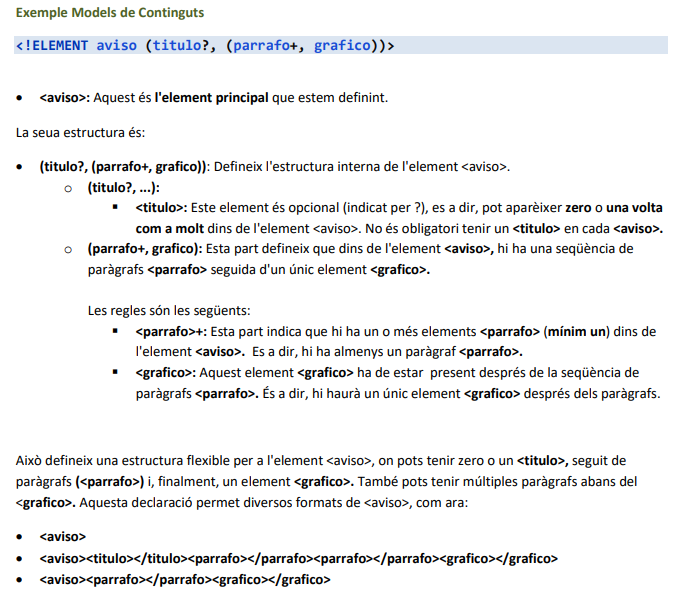
## Tipos de contenidos

* Texto (#PCDATA)
* EMPTY, no puede tener contenido.
* ANY, puede tener cualquier contenido
* Mixted, texto y otros elementos.
* Tipos de elementos, solo pueden tener subelementos.

## Modelo de contenidos

* Identificador general: <!ELEMENT aviso (párrafo)> aviso solo puede contener un solo párrafo.
* Secuencia: se genera mediante “,” <!ELEMENT aviso (párrafo, titulo)> aviso tiene que contener un párrafo seguido de un titulo.
* Opción: se utiliza la “|” en vez de la “,” para dar opción a una cosa o la otra. No hay limite de opciones.

## Interfaz de usuario gráfica, Texto Descripción generada automáticamenteQuantificadores



# Atributos

Son elementos clave que se pueden utilizar para proporcionar información adicional.

* Van insertados en los elementos.
* Están formados por parejas nombre-valor.
* Los valores van entre comillas.
* Pueden ser obligatorios, opcionales, fijos…
* Aceptan valores de texto, números, listas de posibles valores etc.
* Permiten añadir información adicional a los elementos de un documento.
* No pueden contener sub-atributos.
* Se utilizan para añadir información corta, sencilla y desestructurada.
* Solo se pueden especificar una vez.
* Pueden aparecer en cualquier orden.

## Declaracion de atributos en dtd

<!ATTLIST 1nombreElemento 2nombreAtributo 3tipoAtributo 4valorInicialAtributo>

1. Identificador del elemento al que se aplica el atributo.
2. Nombre del atributo.
3. Tipo de atributo.
4. Valor por defecto del atributo.

## Tipos de atributos

## Valores de atributos

* Los valores de los atributos pueden ser literales o referencias.
* En caso de literales los valores tienen que coincidir exactamente con los especificades en ENUMERATS o ser de tipo CDATA.
* Las referencias se utilizan para IDREF, IDEREFS y ENTITAT

## Valores predeterminados de atributos

* Puedes especificar un valor predeterminado para un atributo que se utilizara si no se proporciona un valor en el documento XML.
* Valor entre comillas dobles o simples
* Los valores predeterminados pueden ser
* #REQUIRED (obligatorio)
* #IMPLIED (opcional)
* #FIXED (tiene un valor fijo que no se puede cambiar)

# Entidades

Se utilizan para definir los valores que se pueden reutilizar en todo el documento XML. Pueden ser símbolos, fragmentos de texto o incluso caracteres especiales. Sus principales usos son para ficheros, paginas web, imágenes etc.

<!ENTITY

## Uso de las entidades

Las entidades se utilizan poniendo & seguido del nombre de la entidad y luego un punto y coma.

## Ventajas de las entidades

* Facilitan la reutilización de datos.
* Mejoran la legibilidad del documento XML.
* Permiten gestionar y mantener fácilmente el contenido que utilizan en diversos documentos.

## Entidades internas

* Definidas directamente en el DTD del documento XML.
* Se utilizan principalmente para definir caracteres especiales o fragmentos de texto.
* Son siempre analizadas (una vez que se reemplaza la referencia a la entidad por su contenido pasa a ser parte del documento XML y se analiza por el procesador.



## Entidades generales externas analizadas

Las entidades externas se definen en un fichero separado y se pueden utilizar en diferentes documentos XML.

## Entidades no analizadas

* Si el contenido de la entidad es binario, no se tendría que interpretar como si fuera texto XML.
* Siempre son entidades generales y externas.

# Ejemplo DTD

