

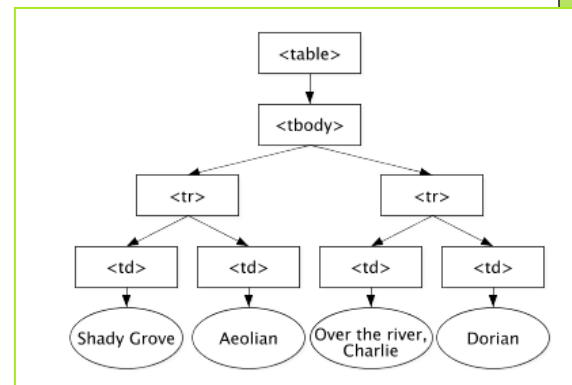
Unidad 1 XML. Iniciación

Módulo: Lenguaje de Marcas y Sistemas de Gestión de Información
Ciclo: Desarrollo de Aplicaciones Web

1.- XML Introducción

◉ Extensible MarkupLanguage = Lenguaje de marcas extensible

- ◉ Deriva del original SGML
- ◉ Principales **aplicaciones**:
 - ◉ Gestión, visualización y organización de datos
- ◉ Lenguaje de visualización XSL
- ◉ El modelo de objeto documento DOM



◉ Problemas principales HTML

- ◉ Mayoría de etiquetas sirven para indicar formato
 - ◉ No para indicar tipo de contenido que se tiene
- ◉ Lenguaje rígido, no extensible
 - ◉ No se pueden añadir etiquetas porque navegador no interpreta

```
<table>
<tbody>
  <tr>
    <td>Shady Grove</td>
    <td>Aeolian</td>
  </tr>
  <tr>
    <td>Sobre el río, Charlie</td>
    <td>Dorian</td>
  </tr>
</tbody>
</table>
```

◉ Objetivos XML

- Similar al HTML
- Extensible
- Reglas concisas, estrictas y fáciles
- Multiplataforma
- Fácil de leer por humanos
- Fácil crear procesador XML (**parsers**)

1.1- Lenguajes basados en XML

- **RSS (*Really Simple Syndication*)**
 - Para producir contenidos sindicables. Por ej. noticias
- **Atom:** Para distribuir información desde una web
 - Muchas veces complementa a RSS
- **ePUB:** Formato de libro digital (estándar de facto)
 - Archivo comprimido (ZIP) que contiene 3 documentos XML que especifican la estructura y contenido del documento
- **SOAP (Simple Object Access Protocol)**
 - Protocolo estándar de comunicación para comunicar con Servicios Web
- **SVG (Scalable Vector Graphics):** Gráfico de vectores escalable
 - Definir imágenes vectoriales para publicarse en páginas web
- **VoiceXML:** Para representar diálogos vocales
- **WSDL (Web Services Description Language)**
 - Lenguaje para definir Servicios Web
- **XHTML:** Versión del HTML compatible con las normas XML

1.2.- Usos del XML

○ Contenido web

- Para escribir páginas web
- Tiene documento de validación conocido por los navegadores que especifica exactamente qué elementos y de qué forma se pueden utilizar en el documento

○ Intercambio de información entre aplicaciones

- Fichero de texto plano: No requiere SW especial para leer su contenido

○ Computación distribuida

- Intercambiar información entre PCs a través de las redes
- No puede contener virus o SW espía
 - Máquinas enrutadoras encaminan XML sin problema al ser texto

○ Generar Información empresarial

- Facilidad de estructurar los datos
- Fácil de dar salida por pantalla o impresión

1.3.- Tecnologías relacionadas con XML

- ◉ **DTD (Document Type Definition):** Definición de tipo de documento
 - ◉ Lenguaje para especificar documentos cuyas reglas han de cumplir los XML a los que se asocien.
 - ◉ Crear **documentos de validación** para archivos XML
- ◉ **XML Schema:** Misma función que DTD pero con sintaxis 100% XML
- ◉ **Relax NG:** Alternativa al DTD y XML Schema con un formato más sencillo
- ◉ **Namespacing:** Espacios de nombres
 - ◉ Permite conseguir nombres únicos dentro de documentos XML
- ◉ **XPath:** Lenguaje de consulta para seleccionar /acceder a partes de un XML
- ◉ **CSS (Cascade StyleSheet):** Hojas de estilo en cascada
 - ◉ Dan formato a un documento XML o HTML
- ◉ **XSLT:** Para lo mismo que CSS con más posibilidades
- ◉ **XQuery:** Para consultar datos de documentos XML como una BD
- ◉ **DOM (Document Object Model):** Para acceder a estructura jerárquica del documento (por ej. usarla dentro de lenguaje de programación)

1.4.- SW para producir XML

- **Editor** de texto:

- Plano

- Bloc de notas de Windows
 - Vi para Linux

- Colorear el código

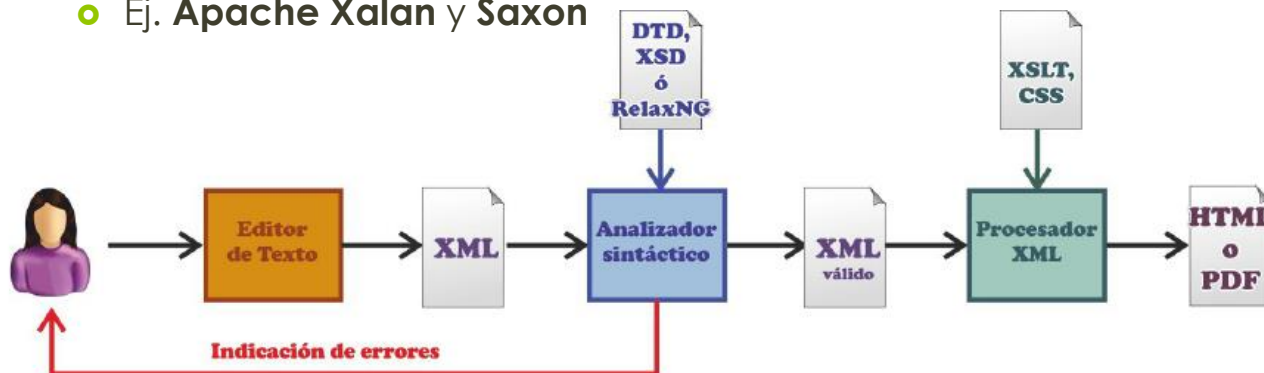
- Notepad++, Scite (Windows)
 - Gedit, Kate, Scribes (Linux)

- **Analizador sintáctico o Parser:** Corrector que entiende XML

- Ej. Apache Xerces

- **Procesador XML** = Navegador (Firefox, Opera...)

- Producir un resultado visual del XML
 - Si se aplica formato visual (como los creados mediante **XSL**)
 - SW especial que convierta datos a la forma final visible por el usuario
 - Ej. **Apache Xalan** y **Saxon**



2.- Estructura XML (I)

○ Etiquetas

- Se deciden según nos interese, no hay una serie de etiquetas que se pueden utilizar

○ Elemento= La etiqueta como lo que contiene ésta

○ Normas:

- Las etiquetas sirven para indicar elementos
 - El nombre de la etiqueta se indica entre los símbolos < y >
- Las etiquetas se cierran indicando </ seguido del NOMBRE DE LA ETIQUETA que se está cerrando
- Distingue entre mayúsculas y minúsculas. Etiquetas en minúscula
- Espaciar y tabular las etiquetas a voluntad
 - Elemento interno a otro aparezca con una sangría mayor
- Comentarios se inician con símbolos <!-- terminan con -->
- Según la **W3C** (organismo de estandarización de XML), el texto en XML debe de estar codificado en **Unicode** (normalmente **UTF-8**)

Ejemplo de XML:

```
<persona>  
    <nombre>Jorge</nombre>  
    <apellido>Sánchez</apellido>  
</persona>
```

2.- Estructura XML (II)

- Documentos XML **se dividen en:**
 - **Prólogo:** 1ª zona del documento.
 - Describe qué tipo de documento es. (Similar al head de HTML)
 - Puede contener
 - **Declaración del documento:** Indicar tipo de documento XML que es.
 - **Instrucciones** para el procesamiento del documento
 - **Comentarios**
 - **Indicación del documento DTD, XSD o Relax NG** para comprobar si el mismo es válido según las reglas impuestas por dicho documento.
 - **Indicación de otros documentos que afectan al actual**, como por ej. documentos **XSLT** que permiten especificar la forma en la que el documento se debe mostrar en pantalla.
 - **Elemento raíz:** Cualquier elemento está dentro de él.
 - Elemento obligatorio que se abre tras el prólogo y se debe cerrar justo al final
 - Contiene:
 - **Más elementos**
 - **Atributos**
 - **Texto normal**
 - **Entidades**
 - **Comentarios**

2.- Estructura XML (III)

○ Reglas para los nombres:

- Elementos, atributos,... tienen un nombre = **Identificador**
 1. Deben comenzar por una letra, y después le seguirán más letras, números o el signo de subrayado o guión bajo.
 2. No pueden empezar con la palabra XML ni en mayúsculas ni en minúsculas ni en ninguna combinación de mayúsculas ni minúsculas
 3. Ojo con las mayúsculas y minúsculas ya que distingue entre ellas

○ Elementos del prólogo:

- **Declaración XML:** 1ª línea del documento XML (opcional)

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

- Indica la versión XML y la codificación (utf-8 es la habitual)

○ Instrucciones de procesamiento

- Indican un documento para validar el XML, darle formato... u otras funciones **<?xml-stylesheet type="text/xsl" href="stylesheet.xsl"?>**
- Asocia un documento **xsl** al XML para darle un formato de salida
- Forma en que se muestran los datos por pantalla

2.- Estructura XML (IV)

- **Elementos del prólogo:**

- **Comentarios:** Comienzan con `<!--` y terminan con `-->`

- Para explicar el código
 - No pueden meterse dentro de la etiqueta, ni tampoco puede contener etiquetas tanto de apertura como de cierre

- **Elementos**

- Sirven para dar significado al texto o a otros elementos o también para definir relaciones entre distintos elementos y datos

- No confundir elemento con etiqueta

- El contenido de elemento puede

- Texto
 - Otros elementos
 - Ambas cosas

- Los elementos se deben abrir y cerrar con la etiqueta que sirve para definir el elemento

- Puede haber incluso elementos vacíos que se cierran en la misma etiqueta

`<nombre>Juan</nombre>`

- `<nombre>` Es un etiqueta de apertura
 - `</nombre>` Es una etiqueta de cierre
 - `<nombre>Juan</nombre>` Es un elemento (el elemento *nombre*)
 - Juan es el contenido del elemento

`<casado></casado>`



`<casado />`

2.- Estructura XML (V)

◉ Elementos del prólogo:

◉ Atributos:

- ◉ Se definen dentro de las etiquetas de apertura de los elementos
- ◉ Se indica su nombre seguido del signo = y del valor (entre comillas) que se le da al atributo
- ◉ Un elemento puede contener varios atributos
 - ◉ Orden es intrascendente pero no pueden atributos repetidos
- ◉ El valor debe aparecer encerrado entre comillas simples o dobles
- ◉ Duda: Cuándo usar elementos y cuándo atributos

```
<peso unidad="gramos" precision="0,01">5,73</peso>
```

◉ Pautas de ayuda:

- ◉ Si la información tiene 1 estructura interna es 1 elemento
- ◉ Si contiene gran cantidad de información es 1 elemento
- ◉ Símil: elemento = sustantivo, atributo = adjetivo
- ◉ Información que tenga un procesamiento o presentación complejos es un elemento
- ◉ En caso de duda usar un elemento

```
<persona privacidad="alta" tipo="autor">  
  <nombre>María</nombre>  
  <apellido>García</apellido>  
</persona>
```

```
<ingrediente>  
  <tipo> tomates</tipo>  
  <peso> 5,73 </peso>  
  <unidad> gramos </gramos>  
  <precision> 0,01 </precision>  
</ingrediente>
```

2.- Estructura XML (VI)

◉ Elementos del prólogo:

- ◉ **Texto:** Siempre entre una etiqueta de apertura y una de cierre
 - ◉ Todo texto es parte de un elemento XML
 - ◉ Se puede escribir cualquier carácter Unicode excepto símbolos <, >...
- ◉ **CSDATA:** Marcar texto para que no sea procesado como parte de XML colocándolo dentro del CSDATA

<![CDATA [texto no procesable...]]>

- ◉ Esto permite usar símbolos <,> y no serán considerados parte de etiquetas
- ◉ Colocar dentro código de lenguajes de scripts como **Javascript** para que no sean interpretados como parte del XML
- ◉ **Entidades:** Representan caracteres individuales
 - ◉ Se usan para representar caracteres especiales que empiezan por & seguidos del nombre de la entidad o nº Unicode del carácter
 - ◉ En XML hay definidas 5 entidades:
 - ◉ **>**; Símbolo >
 - ◉ **<**; Símbolo <
 - ◉ **&**; Símbolo &
 - ◉ **"**; Símbolo “
 - ◉ **'**; Símbolo ‘
 - ◉ Representar caracteres mediante entidades con nº **ñ**; letra ñ

2.- Estructura XML (VII)

o Jerarquías en XML

- o Agrupar en subtemas relacionados y no tener toda la información en un único gran grupo
 - o Más comprensible y más accesible
- o Estructura llamada **modelo de objeto**
- o Información se subdivide en objetos y estos se agrupan en una jerarquía
- o **Agrupar en jerarquías o árbol:** Mediante relación padre/hijo o hermano/hermano
 - o Ej. <name> es padre de <first>, que, por su parte, es hijo de <name>.
 - o <first>, <middle> y <last> son todos hermanos, es decir, son todos hijos de <name>
- o Todas las partes del árbol que contiene hijos se denominan **ramas**, mientras que las partes que no tienen hijos se llaman **hojas**.

```
<name>  
  <first> "John" </first>  
  <middle> "Fitzgerald Johansen" </middle>  
  <last> "Doe" </last>  
</name>
```

2.- Estructura XML (VIII)

○ Jerarquías en XML

- Dependiendo del contenido de los elementos se clasifican en:

- **Elementos con contenido de elemento:** Aquellos que poseen hijos que son a su vez otros elementos

- En nuestro ejemplo <name>

- **Elementos de contenido simple:** Aquellos que solo contienen texto

- En nuestro ejemplo <first>, <middle> y <last>

- **Elementos de contenido combinado:**

- Cuando poseen tanto texto como otros elementos
 - En nuestro ejemplo no hay ningún caso.
 - Las relaciones también se definen haciendo analogía del árbol más compleja
 - <doc> es el ancestro de
 - es un descendiente de <doc>



```
<name>
  <first> "John" </first>
  <middle> "Fitzgerald Johansen" </middle>
  <last> "Doe" </last>
</name>
```

```
<doc>
  <parent>
    Thisissome
    <em>text</em>
    in myelement
  </parent>
</doc>'
```

3.- Documentos bien formados (I)

- Sin un DTD no se puede comprobar la **validez de un documento**
 - Sólo se puede comprobar **si está bien formado**
- **Reglas** para que se considere **documento bien formado**:
 1. Dentro del texto **NO** se pueden usar símbolos de mayor (>), menor (<), ampersand (&) ni las comillas simples o dobles.
 - Se deben usar entidades o deben estar incluidos en una sección CDATA
 2. Símbolos de separación de caracteres (espacios en blanco, tabuladores y saltos de línea) se ignoran.
 - Ej. Se ponen 25 espacios en el texto de 1 elemento → Sólo se considera 1 espacio
 - Sí es posible que sean significativos en algunos elementos (si se indica en su documento de validación)
 3. Todos los caracteres del documento XML forman parte del mismo, será **cuestión posterior** si se tienen en cuenta o no para presentar los datos del documento XML en pantalla o impresión
 4. Los elementos deben estar **anidados adecuadamente**, se cierran primero los últimos elementos abiertos
 5. Valores de los atributos deben encerrarse entre comillas simples o dobles

3.- Documentos bien formados (II)

6. El elemento debe tener **etiqueta de fin** o usar la de elemento vacío
7. El elemento debe tener **un único elemento raíz**
8. Todo texto debe estar incluido en un elemento
9. Los nombres de los elementos comienzan con letras y pueden ir seguidos de letras, n°s, guiones o puntos
 - o Guiones y puntos no recomendables
 - o Los nombres de los elementos NO pueden comenzar con el texto **xml** tanto en minúsculas como en mayúsculas o combinando ambas
- o **El propio navegador** indica si el **XML está bien formado**.
 - o Si escribimos con un editor sin la línea de DOCTYPE que es para llamar a la validación y lo abrimos con un navegador:
 - o Error: Cambiar el nombre de cierre de una etiqueta

