



# **XML**

## **Unidad 1**

*Lenguajes de Marcas y Sistemas de Gestión de Información*



1. **Introducción al lenguaje XML**
2. **Conceptos Básicos del XML**
3. **La Sintaxis XML**
  - i. **La primera línea del documento**
  - ii. **Los elementos**
  - iii. **Los atributos**
  - iv. **La raíz de un documento**
  - v. **Los comentarios**
  - vi. **Los espacios de nombres**

# + 1. INTRODUCCIÓN AL LENGUAJE XML

Compromiso entre la simplicidad de HTML y la complejidad de SGML. El SGML light que crearon fue bautizado primero como *MGML* (*Minimal Generalized Markup Language*), *SLIM* (*Structured Language for Internet Markup*), *MAGMA* (*Minimal Architecture for Generalized Markup Applications*) y luego finalmente *XML* (*eXtensible Markup Language*).

XML se ajusta de manera más eficaz a entornos complejos que HTML. A diferencia de HTML, el XML **separa los datos de su formato**, y su principal propósito es **almacenar la información** y no mostrarla. Veremos que XML incluye un número de archivos, cada uno con su propósito.

# + 1. INTRODUCCIÓN AL LENGUAJE XML

La gran preocupación en la década de los 80 era **separar el contenido** de un documento de su **formato**. Sin embargo, los documentos así creados estaban destinados a imprimirse.

Desde 1992 y con la llegada de la Web y HTML, los documentos electrónicos de hipertexto se han publicado de manera extensa. La disponibilidad de motores de búsqueda ha demostrado la eficacia de la aplicación de tales documentos.

→ SGML era demasiado complejo y no apto para la Web y el HTML no era muy escalable. El W3C quería diseñar un lenguaje que poseyera la **facilidad de implementación de HTML** y la **riqueza semántica del SGML**, naciendo el XML.

## + 2. CONCEPTOS BÁSICOS DEL XML

A diferencia de HTML, el XML no está limitado por un etiquetado predefinido. Dentro de un documento XML, tiene **total libertad para definir sus propias etiquetas**.

Un documento HTML es un documento de texto compuesto de parejas de “etiqueta de apertura/etiqueta de cierre” que describen una función en la organización del documento. Los documentos HTML se pueden editar con un simple editor de texto y se interpretan en los navegadores web.



## 2. CONCEPTOS BÁSICOS DEL XML: MI PRIMER XML

En un documento XML puede crear sus propias etiquetas de descripción:

```
<nombreapellido>Enrique GC</nombreapellido>
```

Esta línea es sintácticamente correcta en XML, pero no se puede utilizar en un documento HTML. Los bloques creados pueden contener nuevos bloques de información:

```
<identidad>
  <apellido>GC</apellido>
  <nombre>Enrique</nombre>
</identidad>
```

Un documento XML es un archivo de texto con la extensión “*.xml*” que puede editar en cualquier editor de texto y que interpreta el navegador web. Se compone de **etiquetas personalizables**. Sin embargo, a diferencia de HTML, el **resultado puede reutilizarse**.



## 2. CONCEPTOS BÁSICOS DEL XML:

### MI PRIMER XML

7

```
<?xml version="1.0" ?>
```

```
<tarjetadevisita>
```

```
    <apellido>Gonzalez</apellido>
```

```
    <nombre>Enrique</nombre>
```

```
    <empresa/>
```

```
    <puesto>Profesor</puesto>
```

```
    <direccion>
```

```
        <numero>5</numero>
```

```
        <calle>La Fuente</calle>
```

```
        <codigopostal>04409</codigopostal>
```

```
        <localidad>Alicun</localidad>
```

```
    </direccion>
```

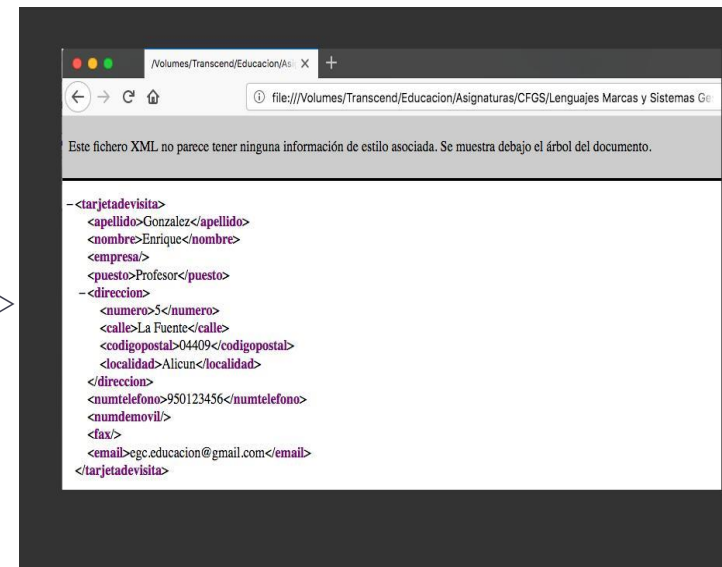
```
    <numtelefono>950123456</numtelefono>
```

```
    <numdemovil/>
```

```
    <fax/>
```

```
    <email>egc.educacion@gmail.com</email>
```

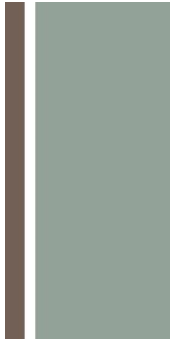
```
</tarjetadevisita>
```





## 2. CONCEPTOS BÁSICOS DEL XML:

### VENTAJAS



El XML es una alternativa eficaz a HTML para la adaptación de nuevos entornos más complejos (comercio electrónico, mensajería, etc...).

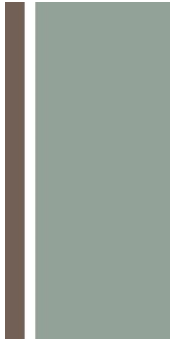
El XML busca **disociar los datos de su presentación**. El HTML ha tenido una clara evolución gracias a la creación de hojas de estilo.

Sin embargo, XML disocia más **eficientemente** los datos y su presentación, su propósito principal es **almacenar información** en lugar de mostrarla.





## 2. CONCEPTOS BÁSICOS DEL XML: ARCHIVOS



El lenguaje XML incluye varios archivos, cada uno con una función específica. El resultado final consiste, de alguna manera, en desarrollar una página HTML.

- El **archivo XML** contiene el conjunto de datos (*archivos.xml*).
- La **DTD (Document Type Definition)** contiene la definición de las características de los elementos que se van a mostrar (*archivos.dtd*).
- Los **archivos CSS (Cascading Style Sheet)** permiten la presentación de los datos (*archivos.css*).
- Los archivos **XSL (XML Stylesheet Language)** son hojas de estilo mucho más potentes que las CSS y mejor adaptadas a XML (*archivos.xsl*).

## + 3. LA SINTAXIS XML

Un archivo XML es un archivo de texto especial compuesto por un determinado número de etiquetas en estructura de árbol. Los elementos del árbol están todos vinculados entre ellos a través de las ramas que se crean.

Si ya conocemos HTML, su diseño es fácil, salvo que hay que tener en cuenta que en XML **todas las etiquetas que se abran deben cerrarse**. Este lenguaje implica un mayor rigor en la escritura del código. Los navegadores han implementado grandes módulos de gestión de errores que ralentizan considerablemente la ejecución de páginas HTML. Si prueba el siguiente ejemplo en un navegador, verá el siguiente error en su página:

```
<?xml version="1.0" ?>
```

```
<A>
```

```
    <B>Etiquetas bien</B>
```

```
    <C>Etiquetas mal
```

```
<A/>
```

**This page contains the following errors:**

error on line 5 at column 5: Opening and ending tag mismatch: C line 0 and A

**Below is a rendering of the page up to the first error.**

Etiquetas bien

## + 3. LA SINTAXIS XML

El archivo XML está organizado según varios componentes que pueden representarse en la forma de un **árbol**. Por ejemplo, el siguiente esquema enumera todos estos componentes.

```
▼<raiz>
  ▼<elemento1>
    ▼<elemento11 atributo="contenido_atributo">
      <elemento111>Contenido Elemento111</elemento111>
      <elemento112>Contenido Elemento112</elemento112>
    </elemento11>
  </elemento1>
  <elemento2>Contenido Elemento2</elemento2>
</raiz>
```

## 3.1. LA PRIMERA LÍNEA DEL DOCUMENTO

La primera línea de un documento XML es el prólogo y contiene la **declaración de la versión** de XML utilizada para la descripción de la información:

```
<?xml version="1.0" ?>
```

El prólogo también permite especificar el **estándar de codificación** de los caracteres utilizados en el documento. La escritura de los caracteres XML cumple con la norma UNICODE. Todos los archivos XML presentados a partir de ahora cumplen con la norma UTF-8.

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

El prólogo también contiene los enlaces a **la información DTD** para la validación y las **hojas de estilo CSS o XSL** para la presentación de los datos.



## 3.2. LOS ELEMENTOS

Los elementos forman la estructura en árbol de un documento XML. El nombre de los elementos se incluye **dentro de las etiquetas**, su valor se coloca entre una etiqueta de apertura y una etiqueta de cierre:

```
<NombreDeElemento>Valor</NombreDeElemento>
```

### **LOS ELEMENTOS VACÍOS**

Un elemento vacío se refiere a un elemento cuyo **valor no se ha rellenado**. Existen dos formas de especificar un elemento de este tipo:

```
<elementovacio></elementovacio>
```

O bien:

```
<elementovacio />
```



## 3.2. LOS ELEMENTOS

### LA SENSIBILIDAD A MAYÚSCULAS Y MINÚSCULAS

Los nombres de las etiquetas de apertura y de cierre deben ser idénticos. Se deben respetar las mayúsculas y minúsculas del nombre:

`<elemento>Contenido</ELEMENTO>`



`<elemento>Contenido</elemento>`



Los nombres de los elementos se escribirán siempre en minúscula.

### LOS CARACTERES NO PERMITIDOS

No se pueden permitir espacios en los nombres de los elementos:

`<nombre elemento></nombre elemento>`



`<nombre-elemento></nombre-elemento>`



### LOS CARACTERES PERMITIDOS

El lenguaje XML permite el uso de algunas caracteres de puntuación dentro de los nombres de los elementos. Puede utilizar, por ejemplo, los siguientes signos: `-'`, `:'`, `\_'`, `.'`. Evidentemente, puede utilizar todos los caracteres alfabéticos (acentuados o no). También pueden utilizarse caracteres numéricos.

## 3.2. LOS ELEMENTOS

### **LA PRIMERA LETRA**

El nombre de un elemento siempre empieza con una letra o uno de los caracteres de puntuación vistos anteriormente:

`<2Marzo2010>Fecha</2Marzo2010>`



`Marzo-2-2010>Fecha</Marzo-2-2010>`



`<\marzo>Mes</\marzo>`



`< <marzo>Mes</marzo>`



### **NOMBRES ESPECIALES**

Los nombres de los elementos que comienzan con XML o 'X' o 'x', 'M' o 'm', 'L' o 'l' deben **evitarse** porque están reservados con fines de normalización del lenguaje.



## 3.3. LOS ATRIBUTOS

Los atributos son **información adicional** relacionada con los elementos XML. La escritura que se utiliza para especificar un atributo de un elemento es la siguiente:

```
<elemento atributo="valor del atributo">Valor del elemento</elemento>
```

En el siguiente ejemplo, el atributo `sexo` permite añadir información al elemento `persona`:

```
<persona sexo="masculino">Enrique</persona>
```

Algunos atributos ya están definidos por el lenguaje. Así, el atributo `xml:lang` permite definir el **lenguaje** utilizado en un archivo XML. El atributo `xml:space` permite **excluir los espacios** múltiples en un nombre.



## + 3.4. LA RAÍZ DE UN DOCUMENTO

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

```
?> <directorio>
```

```
<entrada>
```

```
<nombre>Enrique</nombre>
```

```
<apellidos>GC</apellidos>
```

```
<fechanacimiento />
```

```
<direccion />
```

```
</entrada>
```

```
<entrada>
```

```
<nombre>Amelia</nombre>
```

```
<apellidos>CG</apellidos>
```

```
<fechanacimiento />
```

```
<direccion />
```

```
</entrada>
```

```
<entrada>
```

```
<nombre>María José</nombre>
```

```
<apellidos>PM</apellidos>
```

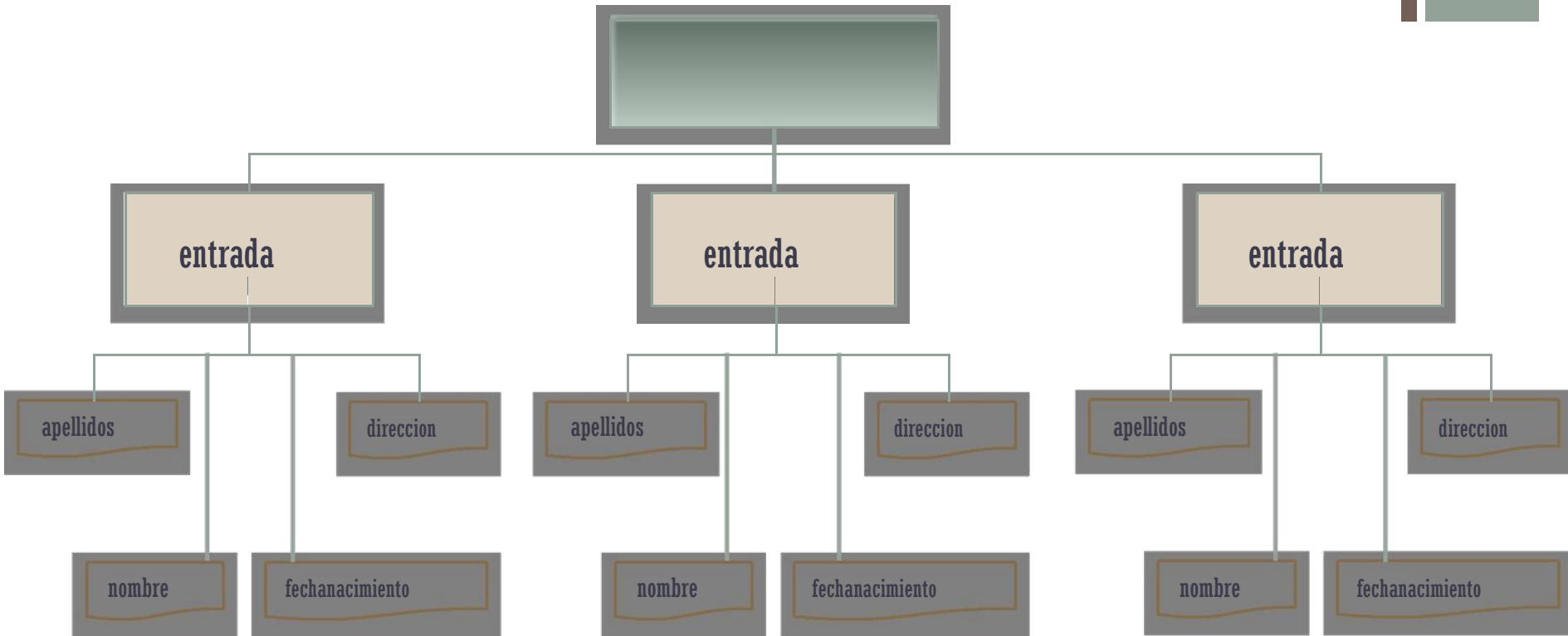
```
<fechanacimiento />
```

```
<direccion />
```

```
</entrada>
```

```
</directorio>
```

## + 3.4. LA RAÍZ DE UN DOCUMENTO





## 3.5. LOS COMENTARIOS

Como en todos los lenguajes, los desarrolladores de XML pueden añadir comentarios de diseño.

Éstos comentarios **no los procesa el compilador**, pero para ello deben incluirse en el documento de la siguiente manera:

```
<!-- Escriba aquí su comentario -->
```

## 3.6. LOS ESPACIOS DE NOMBRES

Los espacios de nombres permiten definir la **pertenencia de los elementos** y los **atributos de un documento XML** a una familia de vocabulario XML. Un nombre de un elemento puede **pertenecer a varias familias** sin que por ello tenga que significar lo mismo.

*Ej:* El elemento `visita` de un sitio web no tendrá la misma información que el elemento `visita` de un museo. Sin embargo, tal vez desee utilizar dos elementos en el mismo documento XML.

En general, los espacios de nombres permiten definir cada nombre de elementos de manera **única**, indexándolos por el nombre de la familia de vocabulario adecuada.

## 3.6. LOS ESPACIOS DE NOMBRES

Los espacios de nombres deben **cargarse previamente** al principio del documento XML antes de poder utilizar los prefijos.

Dentro de un elemento, se hace referencia a un espacio de nombres mediante la siguiente sintaxis:

```
<nombre_elemento xmlns:ALIAS_EDN="URL">
```

ALIAS\_EDN representa el **prefijo** del espacio de nombres que se utilizará en el documento XML y URL es la **dirección del documento** de descripción del espacio de nombres.

De este modo, gracias a los espacios de nombres de nombres, es posible **integrar etiquetas de diversos lenguajes** en un mismo documento XML (HTML, XHTML, SMIL u otros lenguajes de etiquetas).

## 3.6. LOS ESPACIOS DE NOMBRES

En un mismo documento, para utilizar elementos que tengan el **mismo nombre** y que pertenezcan a **dos lenguajes diferentes**, es imprescindible utilizar el concepto **índice (o prefijo)** anteponiendo al nombre del elemento o atributo el alias del espacio de nombres seguido del carácter ":".

El siguiente ejemplo utiliza múltiples espacios de nombres. Tenga en cuenta que su **carga** se configura en la declaración del **elemento raíz**. También puede pertenecer a un espacio de nombres específico:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <ejemplo xmlns:H="http://www.w3.org/REC- html40"
           xmlns:M="http://www.w3.org/REC-MathML">
    <H:b>
      <M:mn>2</M:mn>
    </H:b>
  </ejemplo>
```