

Lenguajes de marcado

Introducción a XML

Aplicaciones Distribuidas
Curso 2025/26

Introducción

- ❑ XML (*eXtensible Markup Language*)
- ❑ Estándar para la representación de datos, con independencia del sistema y del lenguaje de programación.
- ❑ La tecnología XML no establece un lenguaje de modelado de datos, sino que permite la definición de **lenguajes de marcado**:

```
<asignatura curso="2024/25">  
Aplicaciones Distribuidas  
</asignatura>
```

Introducción

❑ Lenguajes de Marcado:

- Lenguajes que incluyen **marcas** que describen cómo procesar el documento
- El marcado es la *información acerca del documento*.
- Ejemplo: **HTML**

```
<html>
<head> <title>Poema</title> </head>
<body lang="es">
  <h1>Alba</h1>
  <h2>Abril de 1915 </h2>
  <h2><em>Granada</em></h2>
  <p>Mi corazón oprimido</p>
  <p>siente junto a la alborada</p>
  <p>el dolor de sus amores</p>
  <p>y el sueño de las distancias.</p>
</body>
</html>
```

Introducción

- ❑ **Marcado generalizado:** **sintaxis común** que facilita la creación de lenguajes de marcas
- ❑ Ejemplo: otras marcas, misma sintaxis

```
<poema fecha="Abril de 1915" lugar="Granada">  
<titulo>Alba</titulo>  
<verso>Mi corazón oprimido</verso>  
<verso>siente junto a la alborada</verso>  
<verso>el dolor de sus amores</verso>  
<verso>y el sueño de las distancias.</verso>  
</poema>
```

Introducción

- ❑ Marcado generalizado, historia:
 - 1970s GML, Generalized Markup Language, en IBM
 - 1986 **SGML** Standard Generalized Markup Language (Estándar ISO)

- ❑ Marcado generalizado:
 - Utilizado para el intercambio de documentos
 - Principio: separar el contenido de la forma de procesarlo
 - Flexible: permite definir lenguajes de marcado específicos para cada aplicación.
 - HTML era un lenguaje de marcado definido en SGML

Introducción

- XML fue desarrollado por:
 - T. Bray (Netscape, W3C, Sun, Google)
 - C. M. Sperberg-McQueen (W3C)
 - J. Paoli (Microsoft)
- **Objetivos:**
 - Crear una versión simplificada de SGML para la web
 - 20% de características de SGML 80% de funcionalidad de SGML
- **Versiones:**
 - Versión 1.0: recomendación del W3C desde 1998.
 - Versión 1.1 aprobada en 2004.

Sintaxis

- La estructura de un documento XML es un **árbol**.

```
<html>
  <head> <title>Poema</title> </head>
  <body lang="es">
    <h1>Alba</h1>
    <h2>Abril de 1915 </h2>
    <h2> <em>Granada</em> </h2>
    <p>Mi corazón oprimido</p>
    <p>siente junto a la alborada</p>
    <p>el dolor de sus amores</p>
    <p>y el sueño de las distancias.</p>
  </body>
</html>
```

Node	Content
▲ [e] html	
▲ [e] head	
[e] title	Poema
▲ [e] body	
@a lang	es
[e] h1	Alba
[e] h2	Abril de 1915
▲ [e] h2	
[e] em	Granada
[e] p	Mi corazón oprimido
[e] p	siente junto a la alborada
[e] p	el dolor de sus amores
[e] p	y el sueño de las distancias.

- Las etiquetas de apertura y cierre definen un **elemento**.
- Los datos entre dichas etiquetas definen el **contenido** de los elementos.

Sintaxis

- ❑ Los elementos se pueden **anidar**, pero no entremezclar.
- ❑ Declaración correcta:

```
<signatura>  
  <nombre> Aplicaciones Distribuidas </nombre>  
</signatura>
```

- ❑ Declaración **incorrecta**:

```
<signatura>  
  <nombre> Aplicaciones Distribuidas </signatura>  
</nombre>
```


Sintaxis

- ❑ **Atributos**: proporcionan información adicional a un elemento.
- ❑ El orden de los atributos no es significativo
- ❑ No puede haber 2 atributos con el mismo nombre para el mismo elemento
- ❑ Las comillas dobles y simples pueden mezclarse

```
<asignatura curso="2024/25">  
Aplicaciones Distribuidas  
</asignatura>
```

Sintaxis

- **Elemento vacío:**

- Las etiquetas no encierran ningún contenido
- Declaración abreviada:

`<validado /> → <validado></validado>`

- Habitualmente, incluyen atributos

```

```

Sintaxis

- Tratamiento de los **espacios en blanco** (tabuladores, saltos de línea):
 - Inicialmente, son de utilidad para formatear el documento
 - Preservar espacios en blanco: atributo **xml:space**
 - Valores: preserve o default

```
<asignatura xml:space="preserve">
```

```
Aplicaciones Distribuidas:
```

```
    Una asignatura
```

```
    del primer cuatrimestre.
```

```
</asignatura>
```

Sintaxis

- **Comentarios** en XML, igual que en HTML:

```
<!-- Comentario -->
```

- Dentro no pueden aparecer los caracteres --
- **Entidades predefinidas** para representar caracteres especiales que no deben ser interpretados como marcas:
 - & → &
 - < → <
 - > → >
 - ' → ‘
 - " → “

Sintaxis

- Las **secciones CDATA** permiten especificar datos, utilizando cualquier carácter, especial o no, sin que se interprete como marcado XML.

```
<ejemplo>
  <![CDATA[
    <html>
      <head><title>Rock &
        Roll</title></head>
  ]]>
</ejemplo>
```

- **Nota:** la cadena "]]>" reservada como delimitador del bloque.

Sintaxis

- Un documento XML tiene dos secciones: Prólogo y Contenido.
- El **prólogo** contiene una etiqueta que identifica el fichero como un documento XML

```
<?xml version="..." encoding="..." standalone="..."?>
```

- **version**: identifica la versión de XML. Es un atributo obligatorio, y acepta los valores “1.0” o “1.1”
- **encoding**: identifica el juego de caracteres usado para codificar los datos. Por defecto es UTF-8. Para el castellano “ISO-8859-1”
- **standalone**: indica si este documento XML puede ser procesado sin recursos externos (esquema o DTD). Si no usa elementos externos, “yes”. Atributo opcional.

Sintaxis

- Definición: un documento XML **está bien formado** si
 - Cumple las reglas sintácticas.
 - Contiene un único elemento raíz.
 - Todas las etiquetas están correctamente anidadas.

Espacios de nombres

□ Motivación:

- Resolver conflictos entre declaraciones con el mismo nombre definidas en distintos tipos de documentos (**esquemas**).

□ Declaración:

- `xmlns:alias=(URN del esquema)`
- El atributo **xmlns** sin alias define el **espacio de nombres por omisión**.

```
<asignatura xmlns:ej1="uri1" xmlns="uri2">  
<ej1:alumno> ...  
</asignatura>
```


Referencias

- Especificaciones:
 - XML: <http://www.w3.org/TR/REC-xml/>

- Tutoriales:
 - <http://www.w3schools.com/xml/>