

# Tema 3. Estándares de anotaciones

# Índice

- **Introducción**
- **Lenguajes de anotaciones. XML**
  - Generalidades
  - Componentes de un documento XML
  - Modelado de datos
  - Fundamentos de las DTD
  - Corrección de un documento XML
- **Estándares de anotaciones en XML**
  - TEI
  - XCES
- **Anotaciones stand-off en XML**

# Introducción

- **Anotación** (metadatos, metainformación):
  - información añadida al documento que no forma parte del mensaje en sí mismo
  - permiten hacer explícita una información que está implícita en el propio texto
  - facilitan el aprovechamiento de los documentos y corpus
- **Corpus anotado:** corpus enriquecido con información lingüística y extralingüística
- Los **corpus anotados** se convierten en recursos útiles para **múltiples usuarios y diversos propósitos**

# Introducción

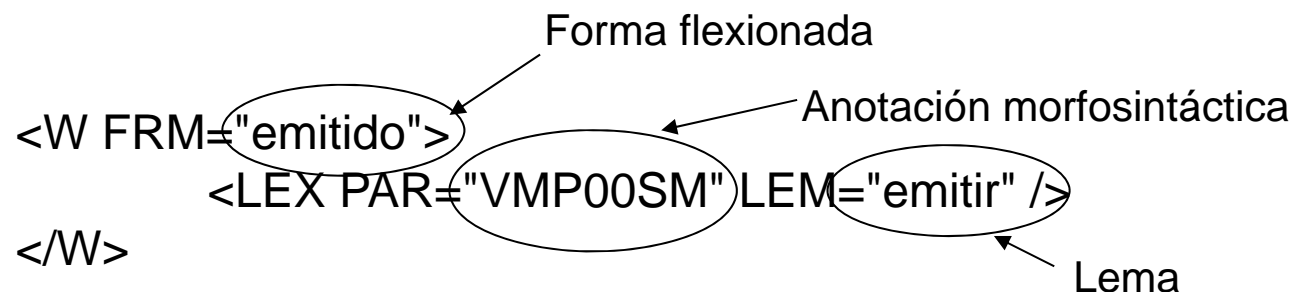
## **Tipos de anotaciones de corpus** [McEnery & Wilson 1996]:

- Anotaciones con **información textual o extratextual** (tipo de corpus, lengua, origen, autor, datos relevantes, etc. )
- Anotaciones con **información ortográfica** (alfabeto distinto del contemplado en el código ASCII de 8 bits, ...)
- Anotaciones con **información lingüística**

# Introducción

Entre las anotaciones con **información lingüística** destacan:

- Anotaciones **morfosintácticas**: asignan a cada unidad léxica una etiqueta o referencia a una entidad indicando su categoría morfosintáctica
- **Lematización**: consiste en la reducción de las palabras flexionadas a su forma base, su lema



# Introducción

- Anotaciones del **discurso**: estructura de los textos, párrafo y oración
- Etiquetado **orientado al problema**: difiere en dos aspectos con respecto a los tipos anteriores:
  - No es exhaustivo en el sentido en que no anota todas las ocurrencias de ciertos elementos (cada palabra, oración, sílaba, ...)
  - Sólo anota los fenómenos relevantes que se estudian (ciertas agrupaciones de palabras, ciertas oraciones, ...)
  - Utiliza un esquema de anotación que no ha sido seleccionado por su amplia cobertura o consenso sino por la relevancia de las distinciones que hace entre los elementos objeto de estudio

# Índice

- Introducción
- **Lenguajes de anotaciones. XML**
  - **Generalidades**
    - Componentes de un documento XML
    - Modelado de datos
    - Fundamentos de las DTD
    - Corrección de un documento XML
- Estándares de anotaciones en XML
- Anotaciones stand-off en XML

# Lenguajes de anotaciones

## **Lenguajes de marcado** (de anotaciones):

- conjunto de reglas que describen cómo deben realizarse anotaciones, bajo qué condiciones se permiten y su significado (éste no siempre)
- permiten hacer explícita la estructura de un documento, su contenido semántico o cualquier otra información lingüística o extralingüística que se quiera hacer patente

`<s>El día <date>21/11/2000 </date> tuvo lugar ... </s>`



# Lenguajes de anotaciones

## Tipos de lenguaje de marcado:

- **Procedimental:** describen operaciones tipográficas
- **Estructural:** describen la estructura lógica de un documento, pero no su tipografía
- **Híbrido:** combinación de ambos

---

**Hojas de estilo:** permiten la “traducción” de las anotaciones estructurales a anotaciones tipográficas

# Lenguajes de anotaciones

## Metalinguajes

- **SGML** (Standard Generalized Markup Language): [ISO 8879:1986]
  - es un metalenguaje que permite definir lenguajes de marcado
  - Su origen es anterior a 1986 (procedía de IBM)
  - La definición de la estructura y el contenido de un tipo (clase) de documento se realiza en una DTD (Document Type Declaration)

# Lenguajes de anotaciones

## ▪ HTML:

- Lenguaje de marcado definido en SGML
- No es un metalenguaje
- **Origen:** 1989 Laboratorio Europeo de Física de Partículas
- **Objetivo inicial:** presentar información estática
- **Limitaciones** relacionadas con:
  - Tratamiento de información dinámica
  - Tiene un número fijo de etiquetas o anotaciones con un significado preestablecido

# Lenguajes de anotaciones. XML

## Metalenguajes

- **XML** (Extensible Markup Language): forma restringida de SGML optimizada para su utilización en Internet (<http://www.w3.org/TR/xml/>)
  - **Origen:** 1996 W3C (consorcio www)
  - **Objetivos iniciales:**
    - Lenguaje estructurado, extensible y que se pueda validar
    - Permitir la transmisión de información realmente estructurada

# Lenguajes de anotaciones. XML

XML Metalenguaje que **describe**:

- una **clase de objetos** de datos (documentos XML)
- parcialmente el comportamiento de los programas que los procesan

# Lenguajes de anotaciones. XML

La especificación 1.1 de XML proporciona:

- su especificación
- los estándares adoptados
  - Unicode e ISO/IEC 10646 para caracteres
  - Internet RFC 3066 [\[IETF RFC 3066\]](#) para identificación de lenguas
  - ISO 639 para códigos de nombres de lenguajes  
<http://www.loc.gov/standards/iso639-2/englangn.html>
  - ISO 3166 para códigos de nombres de países






International  
Organization for  
Standardization

[Home](#) | [Site map](#) | [Abbreviations](#) | [ISO Store](#)  
[Français](#) | [FAQ](#) | [Contact ISO](#) | [My account](#)  
Search  All  ? [Extended Search](#)

[About ISO](#) | [Products and services](#) | [ISO 9000 / 14000](#) | [Standards development](#) | [Communities and markets](#) | [Communication centre](#)



8th Congress on  
Software Quality Management

[Products and services](#)

[ISO Store](#)  
[International Standards](#)  
[Information publications and products](#)  
[Widely used standards](#)  
[Customer services](#)  
[Maintenance Agency for ISO 3166 country codes](#)  
[What's new?](#)  
[ISO 3166 Code lists](#)  
[Updates on ISO 3166](#)  
[Background on ISO 3166](#)  
[ISO 3166 Databases](#)  
[Links](#)  
[Contact](#)

ISO 3166 Code lists  

## English country names and code elements

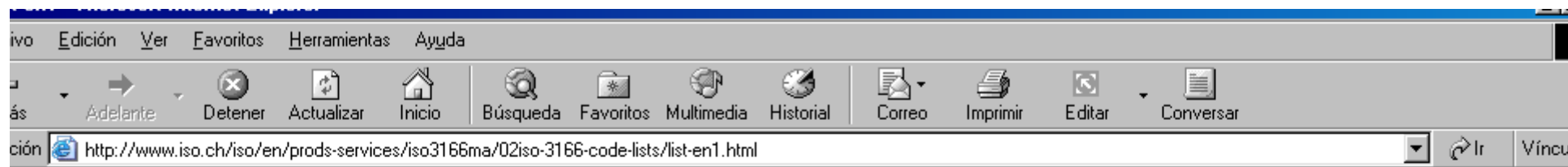
This list states the **country names** (official short names **in English**) in alphabetical order as given in ISO 3166-1 **and** the corresponding **ISO 3166-1-alpha-2 code elements**.

This list is updated whenever a change to the official code list in ISO 3166-1 is effected by the ISO 3166/MA.

It lists 239 official short names and code elements.

[A-F](#) [G-K](#) [L-R](#) [S-Z](#)

AFGHANISTAN	AF
ALBANIA	AL



SAO TOME AND PRINCIPE	ST
SAUDI ARABIA	SA
SENEGAL	SN
SEYCHELLES	SC
SIERRA LEONE	SL
SINGAPORE	SG
SLOVAKIA	SK
SLOVENIA	SI
SOLOMON ISLANDS	SB
SOMALIA	SO
SOUTH AFRICA	ZA
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	GS
SPAIN	ES
SRI LANKA	LK
SUDAN	SD
SURINAME	SR
SVALBARD AND JAN MAYEN	SJ
SWAZILAND	SZ
SWEDEN	SE





# Lenguajes de anotaciones. XML

Distinción entre: Procesador, Aplicación y Analizador

- **Procesador XML:** módulo de software que proporciona acceso al contenido y estructura de un documento XML
- **Aplicación XML:** emplea un procesador XML para acceder al contenido y estructura de un documento XML

Ejemplo: Internet Explorer a partir de la versión 5.0

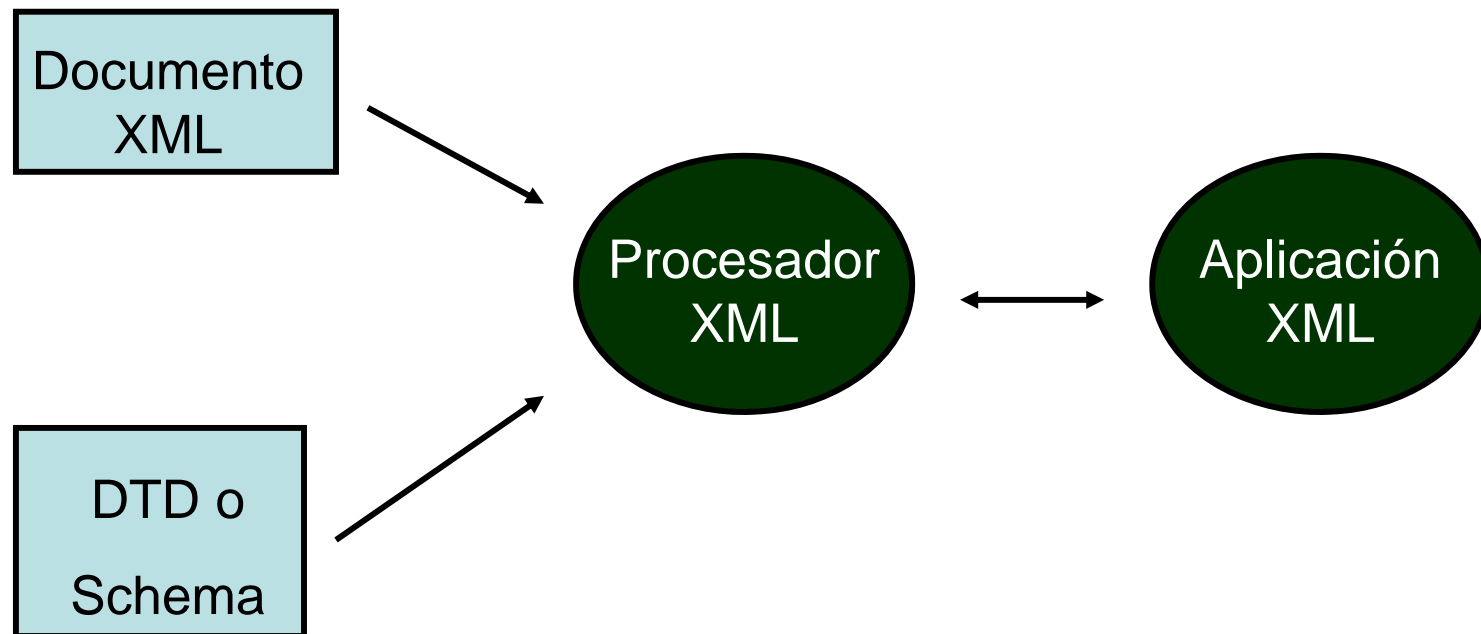
# Lenguajes de anotaciones. XML

Distinción entre: Procesador, Aplicación y Analizador

- **Analizador (parser) XML:** es un componente que requiere el procesador para determinar:
  - La estructura de un documento XML
  - Si es válido o conforme a una DTD o Schema

# Lenguajes de anotaciones. XML

Esquema de una aplicación XML



# Índice

- Introducción
- Lenguajes de anotaciones. XML
  - Generalidades
  - **Componentes de un documento XML**
  - Modelado de datos
  - Fundamentos de las DTD
  - Corrección de un documento XML
- Estándares de anotaciones en XML
- Anotaciones stand-off en XML

# Lenguajes de anotaciones. XML

## Componentes de un documento XML:

- Elementos
- Etiquetas
- Comentarios
- Sección CDATA
- Entidades
- Instrucciones de Procesamiento
- Prólogo

# Lenguajes de anotaciones. XML

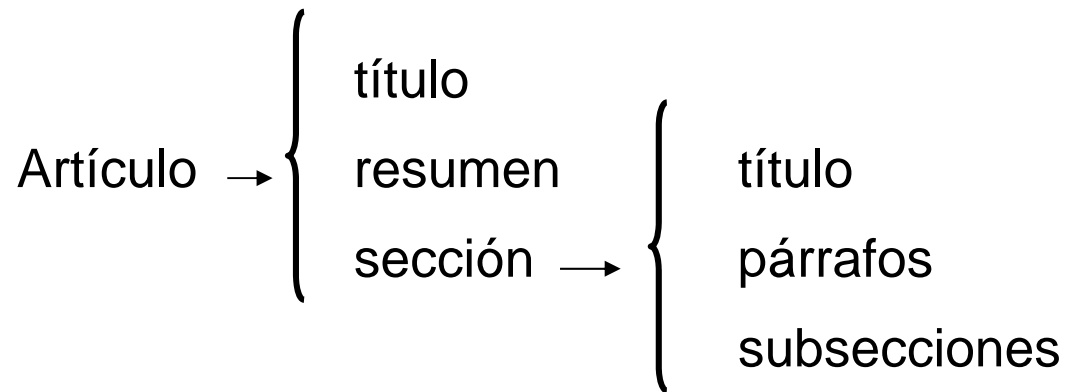
## Componentes de un doc. XML: **Elementos**

- Un elemento es un conjunto de datos del documento delimitado por etiquetas de comienzo y fin de elemento
- Cada elemento representa un **componente lógico** del documento
- Un elemento puede contener otros elementos y normalmente contendrá texto, tablas, etc (***datos carácter***)

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Elementos**

- Ejemplo:



# Lenguajes de anotaciones. XML

Componentes de un doc. XML: **Etiquetas** (nombres de elementos)

- Delimitan los elementos de un documento XML. Su sintaxis es:  
`"<" Identificador-elemento ">"`
- Etiqueta de comienzo de elemento:  
`<identificador-elemento>`
- Etiqueta de fin de elemento:  
`</identificador-elemento>`
- Ejemplo: `<titulo> Introducción </titulo>`



# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Etiquetas** (nombres de elementos)

- Las etiquetas pueden incluir uno o más **atributos**
- Un atributo hace explícita una propiedad del elemento
- Los atributos se expresan en la etiqueta de comienzo y su sintaxis es:

**nombre\_atributo = “valor del atributo”**

- Ejemplos: `<doc lang="en">`    `<p id="1">`

# Lenguajes de anotaciones. XML

Componentes de un doc. XML: **Etiquetas** (nombres de elementos)

- En XML puede haber elementos vacíos
- En XML un elemento vacío consta de:

**<identificador\_elemento atrib1="va1".../>**

- Ejemplo: **<imagen fichero="imagen.gif"/>**

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Comentarios**

- Pueden aparecer en cualquier punto del documento y no forman parte del texto del documento
  - No serán procesados por el analizador
  - Sintaxis: `<!-- comentario -->`
  - Ejemplo: `<!-- esto es un comentario -->`

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Comentarios**

- La cadena “--” no puede aparecer dentro del contenido de un comentario
- Los comentarios pueden ir en cualquier parte de un documento XML excepto:
  - Dentro de las declaraciones
  - Dentro de las etiquetas
  - Dentro de otros comentarios

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Sección CDATA**

- Se utilizan para expresar bloques de texto que contienen caracteres que de otra manera serían reconocidos como etiquetas
- El procesador de XML mostrará su contenido, pero las anotaciones que contenga no serán analizadas
- Sintaxis:

```
<![CDATA[ texto que no se quiere analizar]]>
```

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Sección CDATA**

Ejemplo: `<![CDATA[ <saludo> Hola </saludo> ]]>`

- `<saludo>` y `</saludo>` serían reconocidos como caracteres del texto y no como etiquetas XML
- No se pueden anidar las secciones CDATA

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Entidades**

- Una **entidad** permite asignar un nombre a un subconjunto de datos (texto) y utilizar ese nombre (referencia) para referirnos a dicho subconjunto
- Las entidades pueden aparecer en dos contextos:
  - el documento
  - la DTD

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Entidades**

- Las entidades se pueden clasificar en dos categorías:
  - generales (internas y externas)
  - parámetro
- La marca de comienzo de una referencia a una entidad es “&” y la del final “;”



# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Entidades**

- XML especifica 5 entidades predefinidas:
  - `&amp;` ; para el carácter “&”
  - `&lt;` ; para el carácter “<”
  - `&gt;` ; para el carácter “>”
  - `&apos;` ; para el carácter “ ’ ”
  - `&quot;` ; para el carácter “ “ ”

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Instrucciones de procesamiento**

- Se utilizan para proporcionar información a la aplicación
- Sintaxis:

`<? Instrucción procesamiento ?>`

- Los analizadores XML no hacen nada con las instrucciones de procesamiento, las pasan a la aplicación
- Están prohibidas las que comiencen por XML (salvo la del prólogo)

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Prólogo**

- Los documentos XML pueden empezar con un prólogo en el que se define:
  - Una declaración XML
  - Una declaración de tipo de documento

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Prólogo**

- **La declaración XML:**

- Es una instrucción de procesamiento:

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

- Indica:

- la versión de XML que estamos utilizando (la versión 1.0)
- Información sobre el tipo de codificación de caracteres que estamos utilizando
- En el ejemplo es el código ASCII de 7 bits (subconjunto del código [Unicode](#) denominado UTF-8, los parsers lo manejan por defecto)

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Prólogo**

- La **declaración XML**:

- Otra posibilidad:

```
<?xml version="1.0" encoding="UTF-16"?>
```

- Hay más codificaciones de caracteres además de UTF-8 y UTF-16
- UTF-8 y UTF-16 son las únicas soportadas en todos los procesadores XML

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Prólogo**

- En la **declaración del tipo de documento**, se asocia la DTD respecto a la cual el documento es conforme
- La DTD puede estar en un fichero distinto del documento:

`<!DOCTYPE documento SYSTEM "ejemplo.dtd">` o

`<!DOCTYPE coche SYSTEM`

`"http://www.sitio.com/dtd/coche.dtd">`

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Prólogo**

Con una DTD externa el prólogo ejemplo quedaría:

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

```
<!DOCTYPE coche SYSTEM
```

```
"http://www.sitio.com/dtd/coche.dtd">
```

# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Prólogo**

- La DTD también puede incorporarse en el propio documento XML:

```
<?xml version="1.0"?>
<!DOCTYPE documento [
  <!ELEMENT documento (p | imagen | ejemplo)*>
  <!ELEMENT p (#PCDATA|destacar)*>
  <!ELEMENT destacar (#PCDATA)>
    <!ATTLIST destacar
      importancia CDATA #REQUIRED>
  <!ELEMENT imagen EMPTY>
    <!ATTLIST imagen fichero CDATA #REQUIRED>
  <!ELEMENT ejemplo (#PCDATA)> ]>
```



# Lenguajes de anotaciones. XML

## Componentes de un doc. XML: **Prólogo**

- Ambas partes del prólogo son opcionales, aunque en el caso de incluir ambas la declaración XML tiene que ir antes

# Lenguajes de anotaciones. XML

## Componentes de un documento XML: Codificación de caracteres

- **Referencias a caracteres** (valor numérico)

- Referencia decimal:

**Sintaxis:** &#Num; **Ejemplo:** &#169; (signo copyright)

- Referencia hexadecimal:

**Sintaxis:** &#xNum; **Ejemplo:** &#xA9; (signo copyright)

- Códigos UNICODE: <http://www.unicode.org/charts/>

ivo Edición Ver Favoritos Herramientas Ayuda

ás Adelante Detener Actualizar Inicio Búsqueda Favoritos Multimedia Historial Correo Imprimir Editar Conversar

ción <http://www.unicode.org/charts/PDF/U0080.pdf> Ir Vínculo

158%

5 NEL MW ¥ μ Å Õ å õ  
0085 0095 00A5 00B5 00C5 00D5 00E5 00F5

6 SSA SPA ¡ ¶ Æ Ö æ ö  
0086 0096 00A6 00B6 00C6 00D6 00E6 00F6

7 ESA EPA § • Ç × ç ÷  
0087 0097 00A7 00B7 00C7 00D7 00E7 00F7

8 HTS SOS ¨ , È Ø è ø  
0088 0098 00A8 00B8 00C8 00D8 00E8 00F8

9 HTJ XXX © ¹ É Ù é ù  
0089 0099 00A9 00B9 00C9 00D9 00E9 00F9

A VTS SCI ª ¯ Ê Ú ê ú  
008A 009A 00AA 00BA 00CA 00DA 00EA 00FA

B PLD CSI « » Ë Û ë û  
008B 009B 00AB 00BB 00CB 00DB 00EB 00FB

C PLU ST ¬ ¼ Ì Ü ì ü  
008C 009C 00AC 00BC 00CC 00DC 00EC 00FC

D BL OSC SHY ½ Í Ý í ý

2 de 6 215,9 x 279,4 mm

sto Internet

# Índice

- Introducción
- Lenguajes de anotaciones. XML
  - Generalidades
  - Componentes de un documento XML
  - **Modelado de datos**
  - Fundamentos de las DTD
  - Corrección de un documento XML
- Estándares de anotaciones en XML
- Anotaciones stand-off en XML

# Lenguajes de anotaciones. XML

- **Documento XML:** medio estructurado para almacenar información
- **Esquema (DTD y Schema):**
  - modelo utilizado para describir la estructura de los documentos
  - permiten modelar clases de documentos

# Lenguajes de anotaciones. XML

- Un documento XML debe adherirse a un esquema para ser **válido**
- Sin un esquema un documento puede estar **bien construido** pero no será válido

# Lenguajes de anotaciones. XML

- Un esquema establece restricciones en la estructura del contenido del documento:
  - El modelo de contenido determina el orden y anidación de los elementos
  - Establece los tipos de datos del documento (las DTD no, sólo los Schema)
- Formas de modelar datos en XML: DTD y Schema

# Lenguajes de anotaciones. XML

Diferencias principales entre las DTD y los Schema:

- DTD tienen una sintaxis específica mientras que Schema utiliza sintaxis XML
- Un Schema se puede manipular como cualquier otro documento XML



# Lenguajes de anotaciones. XML

## Diferencias principales entre las DTD y los Schema:

- Un Schema soporta tipos de datos (int, float, boolean, date, ...) las DTD tratan todos los datos como cadenas
- Un Schema soporta la integración de los espacios de nombres permitiendo asociar nodos de un documento con declaraciones de tipo de un esquema
- La DTD solo permite una asociación entre un documento y su DTD

# Índice

- Introducción
- Lenguajes de anotaciones. XML
  - Generalidades
  - Componentes de un documento XML
  - Modelado de datos
  - **Fundamentos de las DTD**
  - Corrección de un documento XML
- Estándares de anotaciones en XML
- Anotaciones stand-off en XML

# Lenguajes de anotaciones. XML

- **DTD (Document Type Definition)** es una formalización de la noción de esquema, tipo o clase de documento
- Una DTD consistirá en una serie de definiciones de tipos de elementos, atributos, entidades y notaciones
- Declara cuales de ellos son legales dentro de un documento y en qué lugar pueden ubicarse

# Lenguajes de anotaciones. XML

- Las DTD son importantes para permitir un procesamiento robusto de los documentos
- Un documento se relaciona con su DTD en la declaración de tipo de documento en el prólogo (DOCTYPE)
- Un documento XML no requiere obligatoriamente tener una DTD o Schema (a diferencia de SGML)

# Lenguajes de anotaciones. XML

Puede haber cuatro tipos de declaraciones en una DTD:

- Declaración de tipo de elemento
- Declaración de atributos
- Declaración de entidades
- Declaración de notación

# Lenguajes de anotaciones. XML

## Declaración de **tipo de elemento**

- Identifican los nombres de los elementos y la naturaleza de su contenido
- Comienzan con "**<!ELEMENT**"
- Sigue el identificador genérico del elemento que se declara
- A continuación tienen una especificación de contenido

# Lenguajes de anotaciones. XML

## Declaración de **tipo de elemento**

Ejemplo: `<!ELEMENT receta (titulo, ingredientes, procedimiento)>`

- el elemento `<receta>` puede contener los elementos `<titulo>`, `<ingredientes>` y `<procedimiento>`
- Todos los elementos deben estar definidos en la DTD

# Lenguajes de anotaciones. XML

## Declaración de **tipo de elemento**

El siguiente documento no sería válido:

```
<receta>  
    <parrafo> ...</parrafo>  
    <titulo>...</titulo>  
    <ingredientes>...</ingredientes>  
    <procedimiento>...</procedimiento>  
</receta>
```

El elemento **<parrafo>** no es un elemento de **<receta>**



# Lenguajes de anotaciones. XML

## Declaración de tipo de elemento

La especificación de contenido puede ser de cuatro tipos:

- **EMPTY**: no tiene contenido pero puede tener atributos

```
<!ELEMENT imagen EMPTY>
```

- Se pueden representar en el documento XML de dos formas:

```
<imagen atrib1= "val1"...> </imagen> 0
```

```
<imagen atrib1= "val1".../>
```

# Lenguajes de anotaciones. XML

## Declaración de **tipo de elemento**

Siguiendo la definición anterior, el siguiente documento:

```
<receta>
  <titulo>...</titulo>
  <ingredientes>...</ingredientes>
  <procedimiento>...</procedimiento>
</receta>
```

Sería válido con respecto a esa DTD

# Lenguajes de anotaciones. XML

## Declaración de tipo de elemento

- **ANY:** Puede tener cualquier contenido (caracteres, otros elementos o mixto)
  - No se suele utilizar, ya que es conveniente estructurar y declarar adecuadamente nuestros documentos XML
  - Ejemplo: `<!ELEMENT detodo ANY>`

# Lenguajes de anotaciones. XML

## Declaración de tipo de elemento

- **Mixto:** Puede tener caracteres o una mezcla de caracteres y sub-elementos
  - Ejemplo de sólo caracteres válidos:
    - `<!ELEMENT enfasis (#PCDATA)>`
    - `#PCDATA` (*datos de carácter analizados*)
  - Ejemplo de caracteres y elementos:
    - `<!ELEMENT parrafo (#PCDATA|enfasis)*>`
  - Si aparece `#PCDATA` en el modelo de contenido debe hacerlo en primer lugar del grupo

# Lenguajes de anotaciones. XML

## Declaración de tipo de elemento

- **Solo elementos:** Sólo puede contener sub-elementos de la especificación de contenido

Ejemplo:

```
<!ELEMENT mensaje (remite,destinatario,texto)>
```

# Lenguajes de anotaciones. XML

Símbolos que pueden aparecer en el modelo de contenido de los elementos

- **Paréntesis “()”**: engloba una secuencia o grupo de sub-elementos

`<!ELEMENT aviso (titulo, parrafo)>`

- **Coma “,”**: denota una secuencia o grupo de sub-elementos

`<!ELEMENT aviso (titulo, parrafo)>`

# Lenguajes de anotaciones. XML

Símbolos que pueden aparecer en el modelo de contenido de los elementos

- **Canalización "|":** separa los elementos de un grupo de alternativas

```
<!ELEMENT aviso (parrafo | grafico)>
```

```
<!ELEMENT aviso (titulo, (parrafo | grafico))>
```

# Lenguajes de anotaciones. XML

Símbolos que pueden aparecer en el modelo de contenido de los elementos: **Indicadores de frecuencia**

- Interrogación “?”: 0 o 1 apariciones
- Asterisco “\*”: de 0 a  $n$  apariciones
- Signo más “+”: de 1 a  $n$  apariciones
- Ejemplo:

```
<!ELEMENT aviso (titulo?, (parrafo+, grafico)*)>
```



# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos

- Los atributos permiten añadir información adicional a los elementos de un documento
- Los atributos no pueden contener sub-atributos
- Se usan para añadir información corta, sencilla y no estructurada
- Un atributo sólo puede aparecer una vez en un elemento y el orden, en caso de existir varios, no es relevante

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos

- Las declaraciones de los atributos empiezan con "**<!ATTLIST**"
- Sigue el identificador del elemento al que se aplica el atributo
- Después el nombre del atributo, su tipo y su valor predeterminado:

```
<!ATTLIST elemento atributo
```

```
tipoAtributo valorPredeterminado>
```

# Lenguajes de anotaciones. XML

Declaración de tipo de atributos: Tipos de **valor predeterminado**:

- **#REQUIRED**: el atributo es obligatorio
- **#IMPLIED**: el atributo es opcional
- **#FIXED *valor***: el atributo tiene un valor fijo (constante)
- ***Valor predeterminado***: el atributo tiene el valor predeterminado

# Lenguajes de anotaciones. XML

Declaración de tipo de atributos. **Tipos de atributos:**

- **CDATA:** datos de caracteres
- **Enumerados:** una serie de valores
- **NOTATION:** una notación declarada en la DTD
- **ENTITY:** una entidad binaria externa
- **ENTITIES:** múltiples entidades binarias externas separadas por espacios en blanco

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos. **Tipos de atributos:**

- **ID:** un identificador único
- **IDREF:** una referencia a un id declarado en la DTD
- **IDREFS:** múltiples referencias a id declarados en la DTD
- **NMTOKEN:** un nombre con sólo caracteres XML válidos (letras, números, puntos, guiones, subrayados y los dos puntos)
- **NMTOKENS:** múltiples nombres con sólo caracteres XML válidos (letras, números, puntos, guiones, subrayados y los dos puntos)

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos

- Ejemplo declaración atributo:

```
<!ATTLIST mensaje fecha CDATA #REQUIRED>
```

Ejemplo uso atributo: `<mensaje fecha="15 de Julio de 1999">`

- Ejemplo declaración atributo:

```
<!ATTLIST mensaje fecha NMTOKEN #REQUIRED>
```

Ejemplo uso atributo: `<mensaje fecha="15-7-1999">`

(sin espacios en el valor del atributo)

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos

- Ejemplo declaración elemento y atributos:

```
<!ELEMENT texto(#PCDATA)>
```

```
    <!ATTLIST texto idioma CDATA #REQUIRED
```

```
        numpalabras CDATA #IMPLIED>
```

el atributo "idioma" es obligatorio, mientras que "numpalabras" es opcional y, si se omite, no toma ningún valor por defecto)

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos

- Ejemplo declaración elemento y atributos:

```
<!ELEMENT mensaje (de, a, texto)>
```

```
<!ATTLIST mensaje prioridad (normal | urgente)
```

```
“normal”>
```

el atributo "prioridad" puede tener el valor "normal" o "urgente", siendo "normal" el valor por defecto si no especificamos el atributo



# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos

- Ejemplo declaración elemento y atributos:

```
<!ELEMENT mensaje (de, a, texto)>
```

```
<!ATTLIST mensaje prioridad (normal | urgente)  
                                #IMPLIED>
```

el atributo "prioridad" puede tener el valor "normal" o "urgente" pero es opcional

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos

- El atributo **NOTATION**: permite al autor declarar que su valor se ajusta a una notación declarada en la DTD

```
<!ATTLIST mensaje fecha NOTATION (ISO-DATE |  
                                EUROPEAN-DATE) #REQUIRED>
```

```
<!ATTLIST imagen formato NOTATION (gif|jpeg)  
                                #REQUIRED>
```

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos. Atributo **ID** e **IDREF**

- ID permite asociar a un atributo un identificador único
- IDREF permite hacer referencia a un atributo ID
- Ejemplo:

```
<!ATTLIST parrafo id ID #REQUIRED>
```

id será un identificador único obligatorio

- Los atributos ID deben declararse como **#REQUIRED** o **#IMPLIED**

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos. Atributo **ID** e **IDREF**

- Ejemplo: implementar un hipervínculo en un documento:

```
<!ELEMENT enlace EMPTY>
```

```
    <!ATTLIST enlace destino IDREF #REQUIRED>
```

```
<!ELEMENT capitulo (parrafo)*>
```

```
    <!ATTLIST capitulo referencia ID #REQUIRED>
```

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos. Atributo **ID** e **IDREF**

- Ejemplo: implementar un hipervínculo en un documento:

```
<enlace destino="seccion-3">
```

haría referencia a:

```
<capitulo referencia="seccion-3">
```

el procesador XML lo podría convertir en un hipervínculo

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos. Atributo **ID** e **IDREF**

- Ejemplo: implementar múltiples referencias (separadas por " "):

```
<!ELEMENT enlace EMPTY>
    <!ATTLIST enlace identificacion ID #REQUIRED>
<!ELEMENT enlace_a_uno (#PCDATA)>
    <!ATTLIST enlace_a_uno enlaza_a IDREF #IMPLIED>
<!ELEMENT enlaces_a_varios (#PCDATA)>
    <!ATTLIST enlaces_a_varios enlaza_a_varios IDREFS #IMPLIED>
```

# Lenguajes de anotaciones. XML

## Declaración de tipo de atributos. Atributo **ID** e **IDREF**

- Ejemplo: implementar múltiples referencias (separadas por " "):

```
<enlace identificacion="seccion-1"/>
```

```
<enlace identificacion="seccion-2"/>
```

```
<enlace identificacion="seccion-3"/>
```

```
<enlace_a_uno enlaza_a="seccion-3"> este elemento enlaza a la sección 3  
</enlace_a_uno>
```

```
<enlaces_a_varios enlaza_a_varios ="seccion-1 seccion-2  
seccion-3"> este elemento enlaza a las tres secciones </enlaces_a_varios>
```

# Lenguajes de anotaciones. XML

## Declaración de entidades

- **Entidades generales:** se usan dentro del contenido del documento
- **Entidades parámetro:** se usan sólo dentro de la DTD



# Lenguajes de anotaciones. XML

## Declaración de entidades generales

- Son entidades analizadas sintácticamente
- Representan y pueden sustituir a una cadena de texto
- Deben ser declaradas:

```
<!ENTITY nombreEntidad definicionEntidad>
```

Ejemplo:   <!ENTITY   uned   "Universidad   Nacional   de  
Educación a Distancia">

# Lenguajes de anotaciones. XML

## Declaración de entidades generales

- Se usan en el contenido del documento

`&nombreEntidad;`

- Ejemplo de declaración:

```
<!DOCTYPE texto [...
```

```
    <!ENTITY uned "Universidad Nacional de Educación a  
                Distancia"> ... ]>
```

- Ejemplo de uso:

```
<texto><titulo> La &uned; </titulo> ...</texto>
```

# Lenguajes de anotaciones. XML

## Declaración de entidades generales

- Las **entidades generales** pueden ser **externas**
- Obtienen su contenido en cualquier otro sitio del sistema (un archivo, una página web, ...)

# Lenguajes de anotaciones. XML

## Declaración de entidades generales externas

Pueden estar:

- Analizadas sintácticamente: contenido XML
- No analizadas sintácticamente:
  - Contenido no XML (texto, datos binarios, ...)
  - Llevan la palabra reservada NDATA en su declaración de entidad

# Lenguajes de anotaciones. XML

## Declaración de entidades generales externas analizadas

- Se utiliza la palabra reservada **SYSTEM** para identificar un archivo del sistema local o de una red
- Se utiliza la palabra reservada **PUBLIC** para identificar un archivo de dominio público (con acceso público)
- Se suele usar **SYSTEM**

# Lenguajes de anotaciones. XML

## Declaración de entidades generales externas analizadas

- La palabra `SYSTEM` seguida de un URI (*Universal Resource Identifier*):

```
<!ENTITY intro SYSTEM  
    "http://www.miservidor.com/intro.xml">
```

- La palabra `SYSTEM` seguida de una ubicación local:

```
<!ENTITY intro SYSTEM "cap1.xml">
```

# Lenguajes de anotaciones. XML

## Declaración de entidades generales externas analizadas

- Permiten que un documento se forme con referencias a sub-documentos:

```
... <!ENTITY cap1 SYSTEM "cap1.xml">
    <!ENTITY cap2 SYSTEM "cap2.xml">
...
<libro>
    &cap1;
    &cap2;
</libro>
```

# Lenguajes de anotaciones. XML

## Declaración de entidades generales externas no analizadas

- No deben ser interpretadas por el procesador XML como si fuera texto XML
- Siempre son generales y externas
- Se referencian por medio de un atributo de tipo

**ENTITY O ENTITIES**



# Lenguajes de anotaciones. XML

## Declaración de entidades generales externas no analizadas

Ejemplo:

```
[ ...<!ELEMENT persona EMPTY>
    <!ATTLIST persona nombre CDATA #REQUIRED
                foto ENTITY #IMPLIED>
<!ENTITY csfoto SYSTEM "csfoto.jpeg" NDATA JPEG> ...]>
<persona nombre="Clara Sanz" foto="csfoto"/>
```

No  
analizada

No se usa &; para denotarla

# Lenguajes de anotaciones. XML

## Declaración de notaciones

- Describen información necesaria para procesar las entidades externas no analizadas
- Ejemplo: notación que describe el tipo de imagen GIF

```
<!NOTATION GIF SYSTEM "GIF">
```

- Se puede ser más explícito y asociar una aplicación:

```
<!NOTATION GIF SYSTEM "Iexplore.exe">
```

# Lenguajes de anotaciones. XML

## Declaración de notaciones

```
<!NOTATION GIF SYSTEM "Iexplore.exe">
```

La aplicación XML podría usar el programa "Iexplore.exe" para ver imágenes GIF que tengan asociada dicha notación

# Lenguajes de anotaciones. XML

## Referencia a entidades externas no analizadas

```
<!ELEMENT persona EMPTY>
  <!ATTLIST persona nombre CDATA #REQUIRED
                    foto ENTITY #IMPLIED>
<!NOTATION JPEG SYSTEM "Iexplore.exe">
<!ENTITY csfoto SYSTEM "csfoto.jpeg" NDATA JPEG>

  <persona nombre="Clara Sanz" foto="csfoto"/>
```

# Lenguajes de anotaciones. XML

## Declaración de entidades de parámetros

- Solo se usan en la DTD
- Permiten modularizar una DTD
- Ejemplo: una entidad que almacene una lista de sub-elementos que se comparten entre varios elementos
- Sintaxis: `<!ENTITY % nombreEntidad definicionEntidad>`

# Lenguajes de anotaciones. XML

## Declaración de entidades de parámetros. Internas

- Ejemplo:

```
<!ENTITY % dimensiones 'ancho, largo'>
```

en la DTD se podrá hacer referencia a:

```
<!ELEMENT pared (alto, %dimensiones;)>
```

```
<!ELEMENT techo (%dimensiones;)> ...
```

- Un uso inadecuado puede añadir complejidad innecesaria

# Lenguajes de anotaciones. XML

## Declaración de entidades de parámetros. Externas

- Ejemplo:

```
<!DOCTYPE texto[  
    <!ENTITY % elemento-a1 SYSTEM "a1.ent">  
    ...  
    %elemento-a1;  
>
```

# Lenguajes de anotaciones. XML

## DTD internas y externas

- La DTD de un documento puede ser:
  - Interna
  - Externa
  - Combinación de ambas

```
<!DOCTYPE elemRaiz SYSTEM DTDexterna  
[DTDinterna]>
```



# Lenguajes de anotaciones. XML

## DTD internas y externas

- Ejemplo:

```
<!DOCTYPE peliculas SYSTEM "peliculas.dtd" [  
    <!ELEMENT actor (#PCDATA)> ]>
```

Elemento raíz: todos los demás deben estar dentro de él (`peliculas`)

DTD externa: "`peliculas.dtd`"

DTD interna: declaraciones adicionales (elemento `actor`)

- Las declaraciones internas prevalecen a las externas

# Lenguajes de anotaciones. XML

Ejemplo: datos de personas con direcciones e-mail. Contenido de “listin.dtd”

```
<!ELEMENT listin (persona)+>
  <!ELEMENT persona (nombre, email*, relacion?)>
    <!ATTLIST persona id ID #REQUIRED>
    <!ATTLIST persona sexo (hombre | mujer) #IMPLIED>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT email (#PCDATA)>
  <!ELEMENT relacion EMPTY>
    <!ATTLIST relacion amigo-de IDREFS #IMPLIED>
    <!ATTLIST relacion enemigo-de IDREFS #IMPLIED>
```

# Lenguajes de anotaciones. XML

Ejemplo de documento xml válido con la DTD "listin.dtd":

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE listin SYSTEM "listin.dtd">
<listin>
  <persona sexo = "hombre" id = "ricky">
    <nombre>Roberto Casas</nombre>
    <email>ro.casas@direccion.com</email>
    <relacion amigo-de = "leire pepe"/>
  </persona>
  <persona sexo = "mujer" id = "leire">
    <nombre>Leire García</nombre>
    <email>le.gracia@direccion.com</email>
    <email>le.garcia@hotmail.com</email>
    <relacion amigo-de = "leire pepe"/>
  </persona>
  <persona sexo = "hombre" id = "pepe">
    <nombre>José Manzaneda</nombre>
    <email>j.manzaneda@direccion.com</email>
    <email>jman@hotmail.com</email>
    <relacion enemigo-de = "leire" amigo-de = "ricky"/>
  </persona>
</listin>
```

# Índice

- Introducción
- Lenguajes de anotaciones. XML
  - Generalidades
  - Componentes de un documento XML
  - Modelado de datos
  - Fundamentos de las DTD
  - **Corrección de un documento XML**
- Estándares de anotaciones en XML
- Anotaciones stand-off en XML

# Lenguajes de anotaciones. XML

Se pueden distinguir dos nociones de corrección en un documento XML:

- documento **válido** (respecto a una DTD o Schema)
- documento **bien construido**

# Lenguajes de anotaciones. XML

- Un **documento** es **valido** si:
  - Está bien construido
  - El nombre del elemento raíz coincide con el nombre de la declaración de tipo de documento
  - La DTD declara todos los elementos, atributos, notaciones y entidades que se utilicen en el documento
  - El documento cumple con la gramática descrita en la DTD
- La validez la determina un analizador o parser

# Lenguajes de anotaciones. XML

- Un objeto de texto es **un documento XML bien construido** si:
  - Tomado como un todo, cumple la regla denominada **"document"**
  - **Respetar todas las restricciones de buena formación** dadas en la especificación
  - **Cada una de las entidades analizadas** que se referencia directa o indirectamente en el documento **está bien formada**

# Lenguajes de anotaciones. XML

Cumplir la regla "**document**" significa:

- Que contiene uno o más elementos
- Hay un elemento llamado raíz (elemento documento) en el que se encuentran todos los demás
- Todo elemento tiene una etiqueta de inicio y de final o una etiqueta de elemento vacío
- Los elementos delimitados por etiquetas de principio y final se anidan adecuadamente
- Los valores de los atributos van entre comillas dobles o simples (" o ')



# Lenguajes de anotaciones. XML

Ejemplo:

```
Mi documento XML
```

no es un documento XML bien construido: no contiene  
ningún elemento

# Lenguajes de anotaciones. XML

Ejemplo:

```
<p> Mi documento XML </p>
```

sí está bien construido: contiene al menos un elemento, "p".

# Lenguajes de anotaciones. XML

Ejemplo:

```
<p> Mi documento XML </p>  
<p> Mi documento XML </p>
```

no es un documento XML bien construido: sólo puede existir un único elemento raíz

# Lenguajes de anotaciones. XML

Ejemplo:

```
<documento>  
    <p> Mi documento XML </p>  
    <p> Mi documento XML </p>  
</documento>
```

si está bien construido: **documento** es el elemento raíz, es único y no forma parte del contenido de ningún otro elemento

# Lenguajes de anotaciones. XML

Ejemplo:

```
<documento>  
    <p> Mi <elem> documento XML </p> </elem>  
    <p> Mi documento XML </p>  
</documento>
```

es incorrecto: la etiqueta inicio del elemento `<elem>` está dentro del contenido del elemento `<p>`, pero su etiqueta final está fuera

# Lenguajes de anotaciones. XML

Ejemplo:

```
<documento>  
    <p> Mi primer <elem> documento XML </elem> </p>  
    <p> Mi primer documento XML </p>  
</documento>
```

documento bien construido

# Lenguajes de anotaciones. XML

## Ejemplo DTD:

```
<?xml version='1.0' encoding='UTF-8' ?>  
<!ELEMENT listin (persona)+>  
<!ELEMENT persona (nombre , email* , relacion?)>  
<!ATTLIST persona id ID #REQUIRED  
                sexo (hombre | mujer ) #IMPLIED >  
<!ELEMENT nombre (#PCDATA)>  
<!ELEMENT email (#PCDATA)>  
<!ELEMENT relacion EMPTY>  
<!ATTLIST relacion amigo-de IDREFS #IMPLIED  
                enemigo-de IDREFS #IMPLIED >
```

# Ejemplo doc. XML conforme con la DTD anterior:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE listin SYSTEM "file:///C:/DTDs/listin.dtd">
<listin>
  <persona sexo = "hombre" id = "ricky">
    <nombre>Roberto Casas</nombre>
    <email>ro.casas@direccion.com</email>
    <relacion amigo-de = "leire pepe"/>
  </persona>
  <persona sexo = "mujer" id = "leire">
    <nombre>Leire Garc a</nombre>
    <email>le.gracia@direccion.com</email>
    <email>le.garcia@hotmail.com</email>
    <relacion amigo-de = "leire pepe"/>
  </persona>
  <persona sexo = "hombre" id = "pepe">
    <nombre>Jos   Manzaneda</nombre>
    <email>j.manzaneda@direccion.com</email>
    <email>jman@hotmail.com</email>
    <relacion enemigo-de = "leire" amigo-de = "ricky"/>
  </persona>
</listin>
```



# Algunas herramientas y utilidades

- <http://saxon.sourceforge.net/dtdgen.html> para generar DTD a partir de un XML (hay que registrarse)
- <http://matra.sourceforge.net/> generar y visualizar DTD
- [http://www.w3.org/2000/04/schema\\_hack/](http://www.w3.org/2000/04/schema_hack/) convertir una DTD en un XML Schema
- <http://www.stylusstudio.com/dtd.html> generar y convertir DTD (periodo de prueba gratuito)

# Índice

- Introducción
- Lenguajes de anotaciones. XML
- **Estándares de anotaciones en XML**
  - **TEI**
  - XCES
- Anotaciones stand-off en XML

# Estándares de anotaciones en XML

## **Text Encoding Initiative (TEI)** (<http://www.tei-c.org/>)

- Iniciativa surgida en 1987 por un conjunto de asociaciones relacionadas con la Lingüística Computacional
- **Objetivo:** proponer un esquema de anotación (DTD) para representar los rasgos (estructurales, lingüísticos, ...) de un texto que necesiten ser hechos explícitos
- **Ventajas:**
  - facilitar el procesamiento de los textos por diferentes aplicaciones y diversos propósitos
  - facilitar el intercambio de textos

# Estándares de anotaciones en XML

## **Text Encoding Initiative (TEI)**

- Especifica un conjunto concreto de etiquetas con un significado
- El esquema puede ser aplicado a cualquier tipo de texto (literario, administrativo, ...) y cualquier lengua
- La primera versión TEI P1 se publicó en 1990 para SGML
- Actualmente la versión más reciente es TEI P5 con una versión para XML (<http://www.tei-c.org/release/doc/tei-p5-doc/es/html/index.html>)

# Estándares de anotaciones en XML

## Text Encoding Initiative (TEI)

- Lista alfabética de todos los elementos de TEI a través de la que se accede a información de sus características y ejemplos en:

<http://www.tei-c.org/release/doc/tei-p5-doc/es/html/REF-ELEMENTS.html>

- **TEI Lite**: subconjunto de TEI suficiente para buena parte de los textos y aplicaciones

- DTD del TEI Lite XML:

<http://www.tei-c.org/Guidelines/Customization/Lite/>

[http://www.tei-c.org/release/xml/tei/custom/schema/dtd/tei\\_lite.dtd](http://www.tei-c.org/release/xml/tei/custom/schema/dtd/tei_lite.dtd)

# Estándares de anotaciones en XML

## Text Encoding Initiative (TEI)

- Ejemplo de un documento anotado con TEI Lite:
  - <http://lists.xml.org/archives/xml-dev/200003/msg00310.html>
  - Ver documento ejemploTEI\_01-es.xml en el curso virtual

# Índice

- Introducción
- Lenguajes de anotaciones. XML
- **Estándares de anotaciones en XML**
  - TEI
  - **XCES**
- Anotaciones stand-off en XML

# Estándares de anotaciones en XML

## **Corpus Encoding Standard for XML (XCES)**

(<http://www.xces.org/>)

- Iniciativa del grupo de trabajo EAGLES Corpus Encoding Standard (CES) adaptado a XML (original en SGML)
- Es una propuesta de anotaciones para información lingüística



# Estándares de anotaciones en XML

## Corpus Encoding Standard for XML (XCES)

- TEI intenta cubrir un abanico muy amplio de aplicaciones, pero algunas relevantes para la ingeniería lingüística no se cubren
- TEI intenta ser muy flexible y ofrece varias maneras de anotar un mismo fenómeno
- XCES limita el esquema de TEI para:
  - Incluir sólo el subconjunto del tagset de TEI relevante para las aplicaciones que procesan corpus
  - Elegir una entre las posibles opciones de codificación que ofrece TEI para ese tipo de aplicaciones

# Estándares de anotaciones en XML

## **Corpus Encoding Standard for XML (XCES)**

- Particularización de TEI a los siguientes tipos de documentos:
  - Documentos que requieren anotaciones relevantes para aplicaciones de ingeniería lingüística
  - Documentos con anotaciones morfosintácticas que necesitan ser enlazadas con otros datos
  - Documentos que contienen enlaces indicando alineaciones entre partes de documentos diferentes

# Estándares de anotaciones en XML

## Corpus Encoding Standard for XML (XCES)

- Restringe o extiende los atributos legales y obligatorios así como los posibles valores especificados en TEI
- No se renombran elementos de TEI excepto cuando puede haber confusiones
- Los elementos de TEI que contienen “TEI” sí se renombran:
  - **<TEI.2>** se renombra por **<cesDoc>**
  - **<teiCorpus.2>** se renombra por **<cesCorpus>**
  - **<teiHeader>** se renombra por **<cesHeader>**

# Estándares de anotaciones en XML

## Corpus Encoding Standard for XML (XCES)

- CES incluye anotaciones que son el resultado del análisis lingüístico de los textos. Por ejemplo:
  - Información morfológica
  - Información sintáctica (e.g., part of speech (POS), salida de un parser)
  - Alineación de corpus paralelos
  - Anotaciones de prosodia
  - ...
- Más información: <http://www.cs.vassar.edu/CES/>

# Índice

- Introducción
- Lenguajes de anotaciones. XML
- Estándares de anotaciones en XML
  - TEI
  - XCES
- **Anotaciones stand-off en XML**
  - **Introducción**
  - Tecnologías XML para su implementación

# Anotaciones *stand-off* en XML

- En los esquemas de anotación convencionales todas las anotaciones de un corpus están en el mismo fichero
- La manipulación de los textos se realiza mediante módulos que tienen como entrada textos y que producen como salida los textos iniciales más cierta información resultado del análisis realizado
- Los módulos pueden ser encadenados de acuerdo a una jerarquía de ejecución (anal. morfológicos, sintácticos, etc. )

# Anotaciones *stand-off* en XML

Cada módulo:

- procesa el conjunto formado por el texto más las anotaciones (si las hubiera)
- selecciona las que le interesan para el análisis que debe realizar
- la salida producida es el conjunto de entrada al que se le habrán añadido las etiquetas resultado de dicho análisis

# Anotaciones *stand-off* en XML

## *Stand-off:*

- En esta arquitectura las anotaciones no están incluidas en el texto original, sino que se conservan en otros documentos SGML o XML independientes con diferentes DTDs o Schemas
- Los documentos con las anotaciones se relacionan mediante enlaces con los textos originales o con otros documentos de anotaciones
- Permite niveles de anotación solapados que de otra manera no son posibles



# Anotaciones *stand-off* en XML

- Descripción de la arquitectura y un ejemplo en el estándar TEI: <http://www.tei-c.org/Activities/SO/sow06.xml?style=printable>
- Ejemplo de arquitectura stand-off en XCES: enlace de elementos <http://www.cs.vassar.edu/XCES/schema/#standoff>
- Herramienta para convertir documentos del “American National Corpus” de arquitectura stand-off a arquitectura convencional: <http://www.anc.org/tools/index.html>
- Descripción de una posible arquitectura con ejemplos: <http://www.xstandoff.net/description.html>

# Índice

- Introducción
- Lenguajes de anotaciones. XML
- Estándares de anotaciones en XML
  - TEI
  - XCES
- **Anotaciones *stand-off* en XML**
  - Introducción
  - **Tecnologías XML para su implementación**

# Anotaciones stand-off en XML

## Tecnologías XML

- Para implementar la arquitectura stand-off en XML es necesario poder acceder a partes concretas de un documento:
  - Un carácter que ocupa una determinada posición
  - Un elemento con un contenido concreto ...
- Además es necesario poder vincular un elemento o atributo de un documento con una parte concreta de otro documento
- El XML no lo permite, se necesitan las denominadas tecnologías XML que añaden diversas funcionalidades

# Anotaciones stand-off en XML

## Vinculación de documentos XML

- No está definida en la especificación XML
- Se define en la especificación XLink del W3C
- Consta de 3 tecnologías:
  - **XPath**: lenguaje para acceder a partes de un documento XML
  - **XPointer**: lenguaje para acceder a la estructura interna de un documento XML (basada en XPath)
  - **XLink**: construcciones de la vinculación avanzada entre documentos (se apoya en XPointer)

# Anotaciones stand-off en XML

## Vinculación XML



# Anotaciones stand-off en XML

## Vinculación XML. XPath

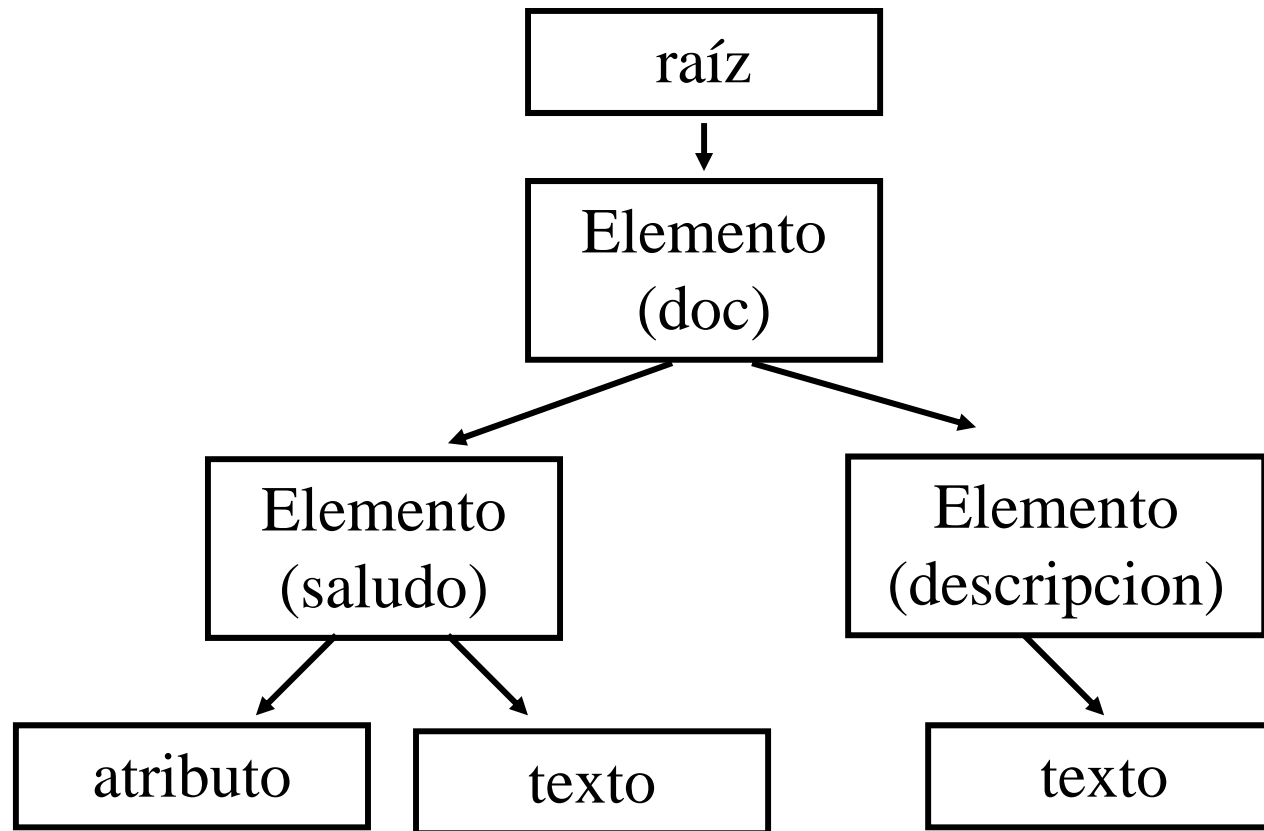
- Es un lenguaje que no usa la sintaxis XML, es un lenguaje es sí mismo
- Proporciona una vía abstracta de dirigirse a partes de documentos XML
- Forma la base del direccionamiento de documentos en XPointer y XSLT (*XML Stylesheets Transformation Language*, o lenguaje de transformación basado en hojas de estilo)
- Se basa en el concepto de notación de ruta, de ahí su nombre

# Anotaciones stand-off en XML

## Vinculación XML. XPath

- Su sintaxis está pensada para ser utilizada en URLs y valores de atributos XML (debe ser concisa)
- Trata el documento como un árbol de nodos
- El árbol lo crea el parser
- La forma en que XPath selecciona partes del documento XML se basa precisamente en la estructura de árbol
- Los "operadores" de que consta este lenguaje permiten recorrer el árbol

```
<doc>
  <saludo tipo= "encuentro"> Hola </saludo>
  <descripcion> encuentro con personas conocidas
</descripcion>
</doc>
```





# Anotaciones stand-off en XML

## Vinculación XML. XPath

- Tipos de nodos para XPath:
  - Raíz: se identifica por “/” (es el nodo raíz no el elemento raíz)
  - Elementos
  - Texto
  - Atributos
  - Espacios de nombres
  - Instrucciones de procesamiento
  - Comentarios

# Anotaciones stand-off en XML

## Vinculación XML. XPath

- Se **basa en expresiones** que permiten recorrer el árbol hasta “llegar” a un nodo determinado
- El resultado de las expresiones es un objeto de datos de uno de los siguientes tipos:
  - **node-set**: un conjunto de nodos
  - **boolean**: un valor true-false
  - **number**: un número en coma flotante
  - **string**: una cadena de texto

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Expresiones.

- Pueden incluir una variedad de operaciones sobre distintos tipos de operandos
- Ejemplos de operandos: *llamadas a funciones* y *location paths* (caminos de localización, localizadores)
- La sintaxis de un *location path* es similar a la usada para describir los caminos en Unix o Linux, pero su significado es diferente

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Expresiones. Ejemplo:

`/libro/capitulo/parrafo`

- hace referencia a TODOS los elementos `parrafo` que cuelguen directamente de TODOS los elementos `capitulo` que cuelguen del elemento `libro` que cuelgue del nodo raíz /
- Una expresión en XPath no devuelve los elementos que cumplen con el patrón que representa dicha expresión, sino la lista de apuntadores a los elementos que encajan en el patrón

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Expresiones. Ejemplo:

```
<libro> <titulo> </titulo> <autor> </autor>  
  <capitulo num="1"> <parrafo> </parrafo> <parrafo> </parrafo>  
</capitulo>  
  <capitulo num="2" public="si"> <parrafo> </parrafo>  
  <parrafo> </parrafo>  
</capitulo>  
  <apendice num="a" public="si"> <parrafo> </parrafo>  
  <parrafo> </parrafo>  
</apendice>  
</libro>
```

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Expresiones.

- Un *location path* siempre tiene un punto de partida llamado **nodo contexto**
- A menos que se indique un camino explícito, se entenderá que el *location path* parte del nodo que en cada momento se esté procesando

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Expresiones. Ejemplo:

`/libro/capitulo/parrafo`

Un evaluador de expresiones Xpath:

- En primer lugar comienza por leer “/” por lo que debe seleccionar el nodo raíz, independientemente del nodo contexto que en ese momento exista
- Cuando el evaluador de XPath localiza el nodo raíz, éste pasa a ser el nodo contexto de dicha expresión

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Predicados.

- Permiten, por ejemplo:
  - seleccionar un nodo que cumple ciertas características
  - seleccionar varios nodos que cumplen con un patrón y con un determinado valor en un atributo
- Los predicados se incluyen dentro de un *location path* utilizando los corchetes:

`/libro/capitulo[@num="1"]/parrafo`

apunta a todos los elementos **parrafo** de todos los elementos **capitulo** que tengan un atributo llamado **num** al cual se le haya asignado el valor "1"



# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Axes

- Permiten realizar una selección de nodos dentro del árbol.

Algunos ejemplos:

***Child:*** es el *axe* utilizado por defecto

- Se corresponde con la barra "/" (la forma larga es: /**child::**)

Ejemplo: `/libro/titulo`

selecciona todos los elementos `titulo` de `libro`

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Axes.

### ***Attribute:***

- su signo es la arroba, @ (su forma larga es: `attribute::`)

Ejemplo: `/libro/capitulo/@num` es equivalente a:

`/libro/capitulo/attribute::num`

selecciona los elementos `capitulo` que posean el atributo `num`

Ejemplo: `/libro/capitulo[@public]/*`

selecciona todos los elementos hijo de los `capitulo` que posean el atributo `public`

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Expresiones. Ejemplo:

`/libro/capitulo[@public]/*`

`<libro> <titulo> </titulo> <autor> </autor>`

`<capitulo num="1"> <parrafo> </parrafo> <parrafo> </parrafo>`

`</capitulo>`

`<capitulo num="2" public="si"> <parrafo> </parrafo>`

`<parrafo> </parrafo>`

`</capitulo>`

`<apendice num="a" public="si"> <parrafo> </parrafo> <parrafo> </parrafo>`

`</apendice>`

`</libro>`

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Expresiones. Ejemplo:

`/libro/capitulo/@num`

`<libro> <titulo> </titulo> <autor> </autor>`

`<capitulo num="1"> <parrafo> </parrafo> <parrafo> </parrafo>  
</capitulo>`

`<capitulo num="2" public="si"> <parrafo> </parrafo> <parrafo> </parrafo>  
</capitulo>`

`<apendice num="a" public="si"> <parrafo> </parrafo> <parrafo> </parrafo>  
</apendice>  
</libro>`

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Axes.

***Descendant:*** selecciona TODOS los nodos que desciendan del conjunto de nodos contexto

- el signo es “//” (su forma larga es: **descendant::**)

Ejemplo: `/libro//parrafo`

**selecciona todos los parrafo de libro**

Ejemplo: `//parrafo//*[@href]`

**selecciona todos los descendientes de parrafo que tienen un atributo href**

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Axes.

***Self:*** selecciona el nodo contexto

- el signo es “.” (su forma larga es: `self::`)

Ejemplo: `./parrafo`

**selecciona todos los `parrafo` descendientes del nodo contexto**

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Nodos *test*

**\*** : devuelve todos los nodos de tipo principal (elemento, atributo o espacio de nombres), pero no nodos de texto, comentarios, y de instrucciones de proceso

Ejemplo: `//capitulo/*`

selecciona todos los nodos descendientes de `capitulo`

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Expresiones.

`//capitulo/*`

`<libro> <titulo> </titulo> <autor> </autor>`

`<capitulo num="1"> <parrafo> </parrafo>`

`<parrafo> </parrafo>`

`</capitulo>`

`<capitulo num="2" public="si"> <parrafo> </parrafo> <parrafo> </parrafo>`

`</capitulo>`

`<apendice num="a" public="si"> <parrafo> </parrafo> <parrafo> </parrafo>`

`</apendice>`

`</libro>`



# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Nodos *test*

***Node()***: devuelve todos los nodos de todos los tipos

Ejemplo: `//capitulo/node()`

selecciona todos los nodos descendientes de `capitulo`

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Expresiones.

`//capitulo/node()`

```
<libro> <titulo> El título </titulo> <autor> El autor </autor>  
  <capitulo num="1"> Capítulo 1 <parrafo> bla, bla </parrafo>  
    <parrafo> bla, bla </parrafo>  
  </capitulo>  
  <capitulo num="2" public="si"> Capítulo 2 <parrafo> bla, bla </parrafo>  
    <parrafo> bla, bla </parrafo>  
  </capitulo>  
  <apendice num="a" public="si"> Apéndice A <parrafo> bla, bla </parrafo>  
    <parrafo> bla, bla </parrafo>  
  </apendice>  
</libro>
```

Párrafo      texto

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Nodos *test*

***text()***: devuelve todos los nodos de tipo texto

***comment()***: devuelve todos los nodos de tipo comentario

***processing-instruction()***: devuelve todos los nodos de tipo  
instrucción del procesamiento

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Más sobre Predicados

- Permiten restringir el conjunto de nodos seleccionados por un *axe* a aquellos que cumplen cierta condición
- La condición es una expresión XPath y se especifica entre corchetes
- Los predicados se pueden suceder uno a otro teniendo el efecto de la operación AND

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Más sobre Predicados

Ejemplo:

```
//capitulo[parrafo/*[@href]][@public='si']
```

es equivalente a:

```
//capitulo[(parrafo/*[@href]) and (@public='si')]
```

Selecciona todos los `capitulo` que tengan un `parrafo` con algún elemento con atributo `href` y que tengan (los `capitulo`) el atributo `public` con valor “si”

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Más sobre Predicados

Ejemplo:

```
//capitulo[parrafo/*[@href]]|//apendice
```

Selecciona todos los `capitulo` que tengan un `parrafo` con algún elemento con atributo `href` o todos los `apendice`

- También se pueden utilizar `or` y `not`

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Ejemplo:

```
//capitulo[parrafo/*[@href]]|//apendice
```

```
<libro> <titulo> </titulo> <autor> </autor>
```

```
  <capitulo num="1"> <parrafo> </parrafo> <parrafo> </parrafo>
```

```
</capitulo>
```

```
  <capitulo num="2" public="si"> <parrafo> </parrafo> <parrafo> </parrafo>
```

```
</capitulo>
```

```
  <apendice num="a" public="si"> <parrafo> </parrafo> <parrafo> </parrafo>
```

```
</apendice>
```

```
</libro>
```

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Más sobre Predicados

Hay funciones que restringen el conjunto de nodos devueltos en una expresión XPath basándose en la posición del elemento devuelto

***position()***: el de la posición indicada

`//capitulo[position()=2]` es equivalente a `//capitulo[2]`

***last()***: el último `//capitulo[not(position()=last())]`

Selecciona todos los capitulo menos el último



# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Más sobre Predicados

***id()***: selecciona elementos con un valor de atributo único **id** igual al indicado

```
id("capitulo_1")/parrafo
```

Selecciona todos los **parrafo** hijo del elemento con

```
id= capitulo_1
```

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Ejemplo:

`//capitulo[2]`

`<libro> <titulo> </titulo> <autor> </autor>`

`<capitulo num="1"> <parrafo> </parrafo> <parrafo> </parrafo>`

`</capitulo>`

`<capitulo num="2" public="si">`

`<parrafo> </parrafo> <parrafo> </parrafo>`

`</capitulo>`

`<apendice num="a" public="si"> <parrafo> </parrafo> <parrafo>`

`</parrafo>`

`</apendice>`

`</libro>`

# Anotaciones stand-off en XML

## Vinculación XML. Xpath. Ejemplo:

```
//capitulo[not(position()=last())]
```

```
<libro> <titulo> </titulo> <autor> </autor>
```

```
<capitulo num="1"> <parrafo> </parrafo> <parrafo> </parrafo>  
</capitulo>
```

```
<capitulo num="2" public="si"> <parrafo> </parrafo> <parrafo>  
  </parrafo>
```

```
</capitulo>
```

```
<apendice num="a" public="si"> <parrafo> </parrafo> <parrafo>  
  </parrafo>
```

```
</apendice>
```

```
</libro>
```

# Anotaciones stand-off en XML

## Vinculación XML. XPointer

- Es una extensión de XPath
- Es un lenguaje que no usa la sintaxis XML
- No es necesario si un vínculo quiere hacer referencia a un documento completo
- Especifica la sintaxis para crear identificadores de fragmentos de un documento
- Permite recorrer la estructura de árbol de un documento XML

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer

- Permite "apuntar" a un punto concreto de un documento
- En HTML lo haríamos:

DocA.htm:

```
<P ID="a1"> Este es el párrafo identificado  
como a1 </P>
```

DocB.htm:

```
<A href= "docA.htm#a1"> Este enlace te llevará  
al párrafo identificado como a1 de docA.htm  
</A>
```

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer

- XPointer va a permitir añadir a una dirección del tipo `http://www.sitio.com/documento.xml`, la coletilla

**`#xpointer(expresión)`**

**expresión:** es una expresión XPath, con algunas propiedades extra que no contempla el propio Xpath

- Una expresión XPointer se añade a un **URI**

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer

- Se pueden concatenar expresiones XPointer que se evalúan de izquierda a derecha mientras devuelvan un conjunto vacío de nodos

Ejemplo:

```
documento.xml#xpointer(/libro/capitulo[@public])xpointer  
(/libro/capitulo[@num="2"])
```

buscaría por en primer lugar el conjunto de nodos delimitado por `/libro/capitulo[@public]` solo en el caso de que no existiese ninguno, buscaría por `/libro/capitulo[@num="2"]`

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer

- Para usar **id()** y evitar que un documento requiera una DTD con un atributo id declarado de tipo único

Ejemplo:

```
documento.xml#xpointer(id("p1"))xpointer(//*[@id="p1"])
```

buscaría en primer lugar **el nodo** con DTD (id único) e id="p1", si no hay DTD o no lo encuentra buscaría elementos (puede haber más de uno) con un atributo id="p1"



# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Puntos y Rangos

**Punto:** una posición en la información XML

**Rango:** selección contigua de toda la información XML que se encuentra entre dos puntos determinados

Ejemplo: con XPath podemos seleccionar perfectamente un nodo principal (elementos, atributos, ...) pero no seleccionar partes de un nodo texto

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Puntos y Rangos

- XPointer considera que existe un punto entre:
  - cualesquiera dos caracteres consecutivos de texto de un documento XML
  - cada par de elementos también consecutivos
- El fragmento de texto que existe entre dos puntos es un rango

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Puntos y Rangos

Ejemplo. En: `<saludo> Hola! </saludo>` hay 13 puntos

1. Antes del nodo `root`
2. Antes del nodo `saludo`
3. Antes del nodo texto `Hola!`
4. Antes del blanco antes de `Hola!` y después de `saludo`
5. Antes de la letra `H`
6. Antes de la letra `o`
7. Antes de la letra `l`
8. Antes de la letra `a`
9. Antes de la letra `!`
10. Después de la letra `!`
11. Después del espacio que divide `!` y `</saludo`
12. Después de `/saludo`
13. Después del nodo `root`.

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Puntos y Rangos

***Point()***: permite usar los puntos en una expresión Xpointer

se le añade un predicado indicando qué punto en concreto se desea seleccionar

Ejemplos: `/point()[position()=5]`

Selecciona el quinto hijo (punto) del nodo root

`/libro/autor/text()/point()[position()=10]`

Selecciona la décima letra de un nodo texto

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Puntos y Rangos

**Rango:** puede describir cualquier zona continua de un documento

- comienza en un determinado punto y finaliza en otro
- cada uno de ellos está determinado por una expresión Xpath

- Para especificar un rango añadimos la función

`/range-to(punto-final)` a la expresión XPath del nodo inicial,  
siendo `punto-final` la expresión XPath del nodo final

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Puntos y Rangos

**Localizaciones:** conjuntos de nodos, más puntos, más rangos

- Algunas funciones de rango:

**`range(location-set)`**: convierte localizaciones en rangos

el rango es el mínimo necesario para cubrir la  
localización entera

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Puntos y Rangos

**Localizaciones:** conjuntos de nodos, más puntos, más rangos

- Algunas funciones de rango:

`start-point(location-set)`: devuelve un conjunto de localizaciones que contiene un punto

Ejemplo: `start-point(//capitulo[1])` devuelve un solo punto

`start-point(//capitulo)` devuelve un conjunto de puntos, cada uno de ellos anterior a `capitulo`

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Puntos y Rangos

**Localizaciones:** conjuntos de nodos, más puntos, más rangos

- Algunas funciones de rango:

`end-point(location-set)`: devuelve justo un conjunto de localizaciones posteriores a cada una de las localizaciones que se le pasan



# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Puntos y Rangos

- Función de rango de cadena:

`string-range(node-set, substring, index,length):`

permite seleccionar cadenas

`xpointer(string-range( /,"noche"))`: Busca todas las

ocurrencias de `noche` en el documento

`xpointer(string-range(//capitulo/titulo,"calle"))`: Busca todas las ocurrencias de `calle` en los `titulo` de los `capitulo` del documento

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Puntos y Rangos

- Función de rango de cadena:

`string-range(node-set, substring, index,length):`

permite seleccionar cadenas

`xpointer(string-range(//capitulo/titulo,"calle",[position()=3])`

Busca sólo la tercera de las ocurrencias de `calle` en los `titulo` de los `capitulo` del documento:

# Anotaciones stand-off en XML

## Vinculación XML. Xpointer. Secuencias de hijos

- Permite recuperar hijos por su posición

Ejemplos:

`documento.xml#/1/3/4` hace referencia al 4º hijo (sea lo que sea), del tercer hijo, del primer hijo que cuelgue del nodo raíz

`#p1/4` hace referencia al cuarto hijo del elemento identificado como p1

# Anotaciones stand-off en XML

## Vinculación XML. XLink

- Tiene sintaxis XML (especificación del W3C <http://www.w3.org/TR/xlink/>)
- Soporta vínculos sencillos (tipo HTML)
- Soporta vínculos extendidos
- Permite vincular 2 documentos a través de un tercero
- Permite especificar la forma de atravesar un vínculo
- Los vínculos pueden residir dentro o fuera de los documentos donde residan los recursos implicados

# Anotaciones stand-off en XML

## Vinculación XML. XLink

- No existen elementos de vinculación predefinidos (no existe <A href=... de HTML)
- Un elemento de vinculación utiliza una construcción denominada *localizador* para conectar recursos
- Define atributos de vinculación estándar

# Anotaciones stand-off en XML

## Vinculación XML. Xlink. Atributos

- **href** Es un localizador de recurso destino (URI, un Xpointer).

Ejemplos:

`"docB.htm" : recurso`

`"docB.htm#xpointer( )" : recurso + XPointer`

- **type** cadena predefinida que determina el tipo de vínculo
- **role** su valor es una cadena de caracteres que aclara el significado o da información adicional sobre el contenido del enlace o vínculo

# Anotaciones stand-off en XML

## Vinculación XML. Xlink. Atributos

- **title**      cadena que sirve de nombre del enlace
- **show**      una cadena predefinida que indica cómo se revela un recurso de destino al usuario
  - **replace**: reemplaza el documento actual por aquel al que apunta el enlace
  - **new**: abre un nuevo navegador con el documento destino
  - **parsed**: el contenido del texto apuntado se incluye en lugar del enlace y se procesa como si fuera parte del mismo documento de origen

## Ejemplo:

```
<doc xmlns:xlink="http://www.w3.org/XML/XLink/1.0">
  ...
  <ver xlink:type="simple"
    xlink:href="http://www.TATE.com/"
    xlink:show="replace">
    Carga la página http://www.TATE.com/
  </ver>
  <nueva-ventana xlink:type="simple"
    xlink:href="http://www.urjc.es/"
    xlink:show="new"> Abre una nueva ventana con la página
    de la URJC
  </nueva-ventana>
  <incluye xlink:type="simple"
    xlink:href="instancia.xml"
    xlink:show="parsed"> Incluye el fichero indicado </incluye>
  ...
</doc>
```



# Anotaciones stand-off en XML

## Vinculación XML. Xlink. Atributos

- **actuate** cadena predefinida que indica cuando se inicia un vínculo (se procede a buscar el destino apuntado)
  - **user**: cuando el usuario pulse o dé alguna orden para seguir el enlace
  - **auto**: el enlace se sigue automáticamente

Ejemplo: redireccionar una página, cargar una imagen o un fichero que debe ser incluido

# Anotaciones stand-off en XML

## Vinculación XML. Xlink. Atributos

- **to**            cadena que identifica el recurso con el que se está vinculando
- **from**        cadena que identifica el recurso desde el que se está vinculando

# Anotaciones stand-off en XML

## Vinculación XML. Xlink.

### Ejemplo. Enlaces simples

```
<doc xmlns:xlink="http://www.w3.org/XML/XLink/1.0">  
    ...  
    <enlace xlink:type="simple"  
            xlink:href="http://www.otraPag.com/"> otra  
página  
    </enlace> ...  
</doc>
```

# Anotaciones stand-off en XML

## Vinculación XML. Xlink.

### Ejemplo. Enlaces simples

```
<doc xmlns:xlink="http://www.w3.org/XML/XLink/1.0">  
    ...  
    <enlace xlink:type="simple"  
        xlink:href=http://www.otraPag.com/  
        xlink:show="replace"> otra          página  
    </enlace> ...  
</doc>
```

# Anotaciones stand-off en XML

## Vinculación XML. Xlink.

### Ejemplo DTD. Enlaces simples

```
<!ELEMENT jugadores ANY>
<!ATTLIST jugadores
  xmlns:xlink CDATA #FIXED "http://www.w3.org/TR/xlink"
  xlink:type (simple|extended|locator|arc) #FIXED "simple"
  xlink:href CDATA #REQUIRED
  xlink:role CDATA #IMPLIED
  xlink:title CDATA #IMPLIED
  xlink:show (new|parsed|replace) "replace"
  xlink:actuate (user|auto) "user">
```

# Anotaciones stand-off en XML

## Vinculación XML. Xlink. Enlaces extendidos

- Pueden tener más de dos recursos que enlazar
- Los recursos se especifican con el elemento `locator`

# Anotaciones stand-off en XML

## Vinculación XML. Xlink. Enlaces extendidos.

Ejemplo:

```
<doc xmlns:xlink="http://www.w3.org/XML/XLink/1.0"> ...  
  <direcciones xlink:type="extended">  
    <locator          xlink:type="locator"          xlink:href="http://w1.com"  
      xlink:show="replace" xlink:actuate="user"/>  
    <locator xlink:type="locator" xlink:href="http://w2.com" xlink:show="new"  
      xlink:actuate="user"/>  
    <locator  xlink:type="locator" xlink:href="http://w3.com"  xlink:show="new"  
      xlink:actuate="auto"/>  
    <locator          xlink:type="locator"          xlink:href="http://w4.com"  
      xlink:show="replace" xlink:actuate="auto"/> las páginas que me interesan  
  </direcciones> ... </doc>
```

# Referencias

- [McEnery & Wilson, 1996] McEnery, T., Wilson, A. *Corpus Linguistics*.  
Edinburgh University Press, 1996.
- [Morrison et al. 2000] “XML al descubierto”. M. Morrison et al. Prentice Hall, 2000.
- [ISO 8879:1986] Information Processing – Text and office Systems – Standard  
Generalized Markup Language (SGML). International Organization for  
Standardization, ISO 8879:1986, 1986.
- [HTML 5 ] HTML 5 Specification. W3C Recommendation.  
<http://www.w3.org/TR/html5/>
- [XML 2.0] Extensible Markup Language (XML) 2.0 W3C Recommendation.  
<http://www.w3.org/TR/xml-c14n2/>
- [XML Path] XML Path Language (XPath) Version 3.0  
<http://www.w3.org/TR/xpath-30/>
- [XML Pointer] W3C XML Pointer, XML Base and XML Linking  
<http://www.w3.org/XML/Linking>
- [XML XLink] W3C XML Xlink 1.1 <http://www.w3.org/TR/xlink11/>