

# XML: The eXtensible Markup Language

Rafael C. Carrasco

Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante

8 de febrero de 2007

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# La importancia de los estándares

## Estándares y metainformación

Aspecto y estructura  
Prospectiva

## Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

## Sintaxis

XMLSchema  
RELAX NG

## Semántica

## Xtandards

XPath  
XQuery  
XQuery  
XLink

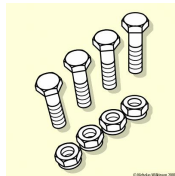
## Procesamiento

SAX Y DOM  
libxml2

XML es un estándar (ISO 8879 simplificado)

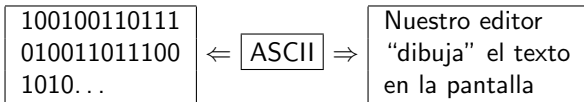
XML, como otros estándares,

- 1 reduce costes de producción
- 2 facilita la preservación a largo plazo.



# Diferencias entre XML y ASCII

**ASCII** (American Standard Code for Information Interchange) es un estándar (desde 1963) para almacenar textos (información) en clave binaria:



Estándares y metainformación

Aspecto y estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Diferencias entre XML y ASCII

## Estándares y metainformación

Aspecto y estructura  
Prospectiva

## Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

## Sintaxis

XMLSchema  
RELAX NG

## Semántica

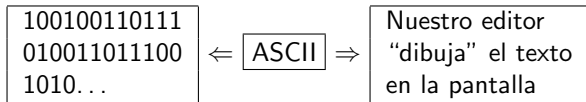
## Xtandards

XPath  
XQuery  
XQuery  
XLink

## Procesamiento

SAX Y DOM  
libxml2

**ASCII** (American Standard Code for Information Interchange) es un estándar (desde 1963) para almacenar textos (información) en clave binaria:



**XML** (eXtensible Markup Language) es un estándar que permite añadir **metainformación** a los textos.

# ¿Qué es la metainformación?

La **metainformación** es información sobre la información, esto es, sobre el contenido. Por ejemplo, una ficha catalográfica es metainformación:

155.2

NUTes Nutlin, Joseph

La estructura de la personalidad / Joseph Nutlin.  
Buenos Aires : Kapelusz, 1973.

237 p. : il.- -(Biblioteca de Psicología Contemporánea; 27)

1.- PSICOLOGIA 2.- PERSONALIDAD

Estándares y metainformación

Aspecto y estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# ¿Donde hay metainformación?

## Estándares y metainformación

- Aspecto y estructura
- Prospectiva

## Especificación

- Estructura lógica
- Estructura física
- Hojas de estilo

## Sintaxis

- XMLSchema
- RELAX NG

## Semántica

## Xtandards

- XPath
- XQuery
- XQuery
- XLink

## Procesamiento

- SAX Y DOM
- libxml2

Tradicionalmente la metainformación era externa (p. ej. ficha catalográfica).

# ¿Donde hay metainformación?

## Estándares y metainformación

Aspecto y estructura  
Prospectiva

## Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

## Sintaxis

XMLSchema  
RELAX NG

## Semántica

## Xtandards

XPath  
XQuery  
XQuery  
XLink

## Procesamiento

SAX Y DOM  
libxml2

Tradicionalmente la metainformación era externa (p. ej. ficha catalográfica).

Pero hay metainformación implícita en:

- 1 el índice de un libro;
- 2 el tipo de letra usado;
- 3 la maquetación.

# ¿Qué metainformación implícita contiene esta imagen?

## Estándares y metainformación

Aspecto y estructura  
Prospectiva

## Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

## Sintaxis

XMLSchema  
RELAX NG

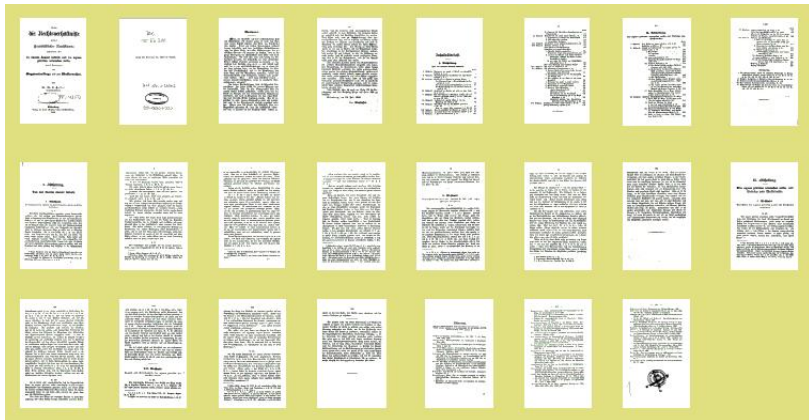
## Semántica

## Xtandards

XPath  
XQuery  
XQuery  
XLink

## Procesamiento

SAX Y DOM  
libxml2





# Textos con XML

XML dice: la metainformación se hace explícita mediante marcas, esto es, texto añadido entre llaves angulares.

```
<title> Doña Perfecta </title>
<author> Benito Pérez Galdós </author>
<text>
    ....
    <note>
        Doña Perfecta, como mujer de su época, ....
    </note>
</text>
```

Estándares y metainformación

Aspecto y estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Textos con XML

## Estándares y metainformación

Aspecto y estructura  
Prospectiva

## Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

## Sintaxis

XMLSchema  
RELAX NG

## Semántica

## Xtandards

XPath  
XQuery  
XQuery  
XLink

## Procesamiento

SAX Y DOM  
libxml2

XML es especialmente adecuado para colecciones de textos, pues permite

- 1 integrar la metainformación en el mismo documento evitando inconsistencias (p.ej., el autor es identificado con el nombre con el que aparece en el libro);

# Textos con XML

## Estándares y metainformación

Aspecto y estructura  
Prospectiva

## Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

## Sintaxis

XMLSchema  
RELAX NG

## Semántica

## Xtandards

XPath  
XQuery  
XQuery  
XLink

## Procesamiento

SAX Y DOM  
libxml2

XML es especialmente adecuado para colecciones de textos, pues permite

- 1 integrar la metainformación en el mismo documento evitando inconsistencias (p.ej., el autor es identificado con el nombre con el que aparece en el libro);
- 2 distinguir qué parte del contenido es promovida a metainformación, facilitando así su explotación (podremos, por ejemplo, localizar los comentarios del editor sobre el personaje.).

# XML no es HTML

XML permite utilizar un vocabulario propio

Por ejemplo:

- author, title, ... para obras literarias;
- calle, código postal, ... para una colección de direcciones.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML no es HTML

XML permite utilizar un vocabulario propio

Por ejemplo:

- author, title, ... para obras literarias;
- calle, código postal, ... para una colección de direcciones.

HTML (el formato de los documentos de internet) tiene un conjunto de etiquetas fijo (head, body, p, ...) que se usan para determinar el aspecto de los documentos de internet.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Relación entre XML, SGML y HTML

## Estándares y metainformación

Aspecto y estructura  
Prospectiva

## Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

## Sintaxis

XMLSchema  
RELAX NG

## Semántica

## Xtandards

XPath  
XQuery  
XQuery  
XLink

## Procesamiento

SAX Y DOM  
libxml2

- XML sigue el estándar **SGML** (Standard Generalized Markup Language, ISO8879).

# Relación entre XML, SGML y HTML

## Estándares y metainformación

Aspecto y estructura  
Prospectiva

## Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

## Sintaxis

XMLSchema  
RELAX NG

## Semántica

## Xtandards

XPath  
XQuery  
XQuery  
XLink

## Procesamiento

SAX Y DOM  
libxml2

- XML sigue el estándar **SGML** (Standard Generalized Markup Language, ISO8879).
- SGML es muy potente pero también muy complejo: no ha sufrido modificaciones desde su creación en 1986.

# Relación entre XML, SGML y HTML

## Estándares y metainformación

Aspecto y estructura  
Prospectiva

## Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

## Sintaxis

XMLSchema  
RELAX NG

## Semántica

## Xtandards

XPath  
XQuery  
XQuery  
XLink

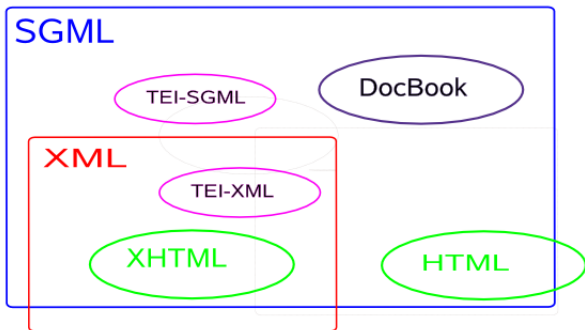
## Procesamiento

SAX Y DOM  
libxml2

- XML sigue el estándar **SGML** (Standard Generalized Markup Language, ISO8879).
- SGML es muy potente pero también muy complejo: no ha sufrido modificaciones desde su creación en 1986.
- HTML sigue la norma SGML pero no sus principios. **XHTML** es HTML conforme a XML.



# Relación entre XML, SGML y HTML



Estándares y metainformación

Aspecto y estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML no es metainformación estructurada

## Estándares y metainformación

Aspecto y estructura  
Prospectiva

## Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

## Sintaxis

XMLSchema  
RELAX NG

## Semántica

## Xtandards

XPath  
XQuery  
XQuery  
XLink

## Procesamiento

SAX Y DOM  
libxml2

Las *bases de datos* funcionan como las fichas catalográficas: contienen información muy estructurada por lo que

- se deben diseñar cuidadosamente desde el principio;
- es difícil integrar nuevos registros;
- no permiten manejar con comodidad textos de longitud variable.

XML ocupa un lugar intermedio entre el texto plano y las bases de datos

# XML no determina el aspecto

Una de las virtudes de XML es la separación entre el diseño lógico y el diseño de presentación (tipografía, gráficos etc).

Formato	Esto es un <code>&lt;b&gt; aviso &lt;/b&gt;</code>	Esto es un <b>aviso</b>
Estructura	Esto es un <code>&lt;warning&gt; aviso &lt;/warning&gt;</code>	Esto es un <b>aviso</b>

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Ventajas del mercado estructural

El mercado estructural permite:

- 1 separar las tareas del editor y del tipógrafo;

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

# Ventajas del mercado estructural

El mercado estructural permite:

- 1 separar las tareas del editor y del tipógrafo;
- 2 realizar búsquedas elaboradas: p. ej. Sevilla en un pie de imagen;

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

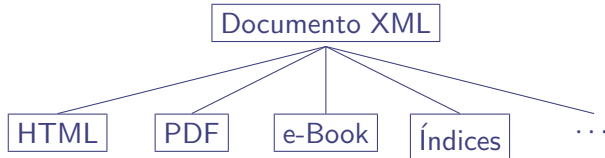
Procesamiento

SAX Y DOM  
libxml2

# Ventajas del marcado estructural

El marcado estructural permite:

- 1 separar las tareas del editor y del tipógrafo;
- 2 realizar búsquedas elaboradas: p. ej. Sevilla en un pie de imagen;
- 3 cambiar el formato o estilo de la presentación para todos los documentos a un tiempo.



Estándares y metainformación

Aspecto y estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Ventajas del marcado estructural

Si incluimos metainformación sobre el idioma del texto, el editor podrá decidir automáticamente la división silábica adecuada según el tamaño de la línea.

La expresión proprietary se suele traducir incorrectamente al castellano como propietario cuando significa “de marca, comercial”.

Estándares y metainformación

Aspecto y estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Ventajas del mercado estructural

## Los formatos WYSIWYG:

- 1 Dificultan integrar documentos con estilos diferentes.
- 2 Permiten proliferación de aspectos que pueden ser ignorados.
- 3 Ocupan más espacio y la información sobre aspecto se repite.
- 4 No permiten explotar la metainformación que el aspecto contiene (almacenada en forma binaria).
- 5 Depende de compañías para recuperar documentos deteriorados o antiguos.

Estándares y metainformación

Aspecto y estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2



# Presente y futuro del XML

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

**Prospectiva**

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## ¿Por qué XML ahora?

# Presente y futuro del XML

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## ¿Por qué XML ahora?

- 1 El interés de establecer sistemas compatibles (estándar) de intercambio de información.

# Presente y futuro del XML

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## ¿Por qué XML ahora?

- 1 El interés de establecer sistemas compatibles (estándar) de intercambio de información.
- 2 El aumento de capacidad de procesamiento de los sistemas actuales.

# Presente y futuro del XML

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## ¿Por qué XML ahora?

- 1 El interés de establecer sistemas compatibles (estándar) de intercambio de información.
- 2 El aumento de capacidad de procesamiento de los sistemas actuales.
- 3 El aumento de la interconectividad (Internet).

# Presente y futuro del XML/2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

**Prospectiva**

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

¿Merece la pena la inversión?

# Presente y futuro del XML/2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

¿Merece la pena la inversión?

1 Es sencillo de aprender.

# Presente y futuro del XML/2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## ¿Merece la pena la inversión?

- 1 Es sencillo de aprender.
- 2 Hay herramientas y programas baratos.

# Presente y futuro del XML/2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## ¿Merece la pena la inversión?

- 1 Es sencillo de aprender.
- 2 Hay herramientas y programas baratos.
- 3 Es legible por humanos.



# Presente y futuro del XML/2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## ¿Merece la pena la inversión?

- 1 Es sencillo de aprender.
- 2 Hay herramientas y programas baratos.
- 3 Es legible por humanos.
- 4 Admite internacionalización (Unicode).

# Presente y futuro del XML/2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
**Prospectiva**

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

## ¿Merece la pena la inversión?

- 1 Es sencillo de aprender.
- 2 Hay herramientas y programas baratos.
- 3 Es legible por humanos.
- 4 Admite internacionalización (Unicode).
- 5 Contribuye a la preservación.

# Presente y futuro del XML/2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
**Prospectiva**

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

## ¿Merece la pena la inversión?

- 1 Es sencillo de aprender.
- 2 Hay herramientas y programas baratos.
- 3 Es legible por humanos.
- 4 Admite internacionalización (Unicode).
- 5 Contribuye a la preservación.

Un ejemplo: colección de referencias bibliográficas.

# Inconvenientes del XML

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

Por contra:

- 1 La información ocupa más espacio.

# Inconvenientes del XML

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

Por contra:

- 1 La información ocupa más espacio.
- 2 Requiere preproceso de los ficheros.

# Inconvenientes del XML

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

Por contra:

- 1 La información ocupa más espacio.
- 2 Requiere preproceso de los ficheros.
- 3 El análisis de árboles es más lento que el de texto plano.

# XML en la actualidad

## El futuro inmediato:

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

**Prospectiva**

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## El futuro inmediato:

- 1 En los *sistemas de información* permite compartir y reutilizar contenidos (por ejemplo, SCORM).



Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## El futuro inmediato:

- 1 En los *sistemas de información* permite compartir y reutilizar contenidos (por ejemplo, SCORM).
- 2 En *e-bussiness* sirve para representar transacciones comerciales.

# XML en la actualidad

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## El futuro inmediato:

- 1 En los *sistemas de información* permite compartir y reutilizar contenidos (por ejemplo, SCORM).
- 2 En *e-bussiness* sirve para representar transacciones comerciales.
- 3 Simplifica comunicación entre sistemas para la *computación distribuida* (SOAP, etc.).

Estándares y  
metainforma-  
ción

Aspecto y  
estructura

Prospectiva

Especificación

Estructura lógica

Estructura física

Hojas de estilo

Sintaxis

XMLSchema

RELAX NG

Semántica

Xtandards

XPath

XQuery

XQuery

XLink

Procesamiento

SAX Y DOM

libxml2

## El futuro inmediato:

- 1 En los *sistemas de información* permite compartir y reutilizar contenidos (por ejemplo, SCORM).
- 2 En *e-bussiness* sirve para representar transacciones comerciales.
- 3 Simplifica comunicación entre sistemas para la *computación distribuida* (SOAP, etc.).
- 4 Permitirá añadir significado a los contenidos y construir la llamada *WWW semántica*.

# XML bien construido

Un documento de XML **bien construido** sólo puede contener:

❶ Texto: `Aquella mañana...`

❷ Pares de etiquetas:

```
<book>...</book>
```

❸ Instrucciones para el procesador:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

❹ Comentarios y declaraciones:

```
<!--Esto es un comentario-->  
<!ENTITY xml "eXtensible Markup Language">
```

❺ Referencias a entidades predefinidas:

```
Welcome to \&xml;.
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML bien construido

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Además, en el XML bien construido:

- 1 Toda etiqueta abierta debe cerrarse.

# XML bien construido

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Además, en el XML bien construido:

- 1 Toda etiqueta abierta debe cerrarse.
- 2 Se distingue nombres en minúscula y mayúscula.

# XML bien construido

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Además, en el XML bien construido:

- 1 Toda etiqueta abierta debe cerrarse.
- 2 Se distingue nombres en minúscula y mayúscula.
- 3 Se debe respetar el anidamiento.

# XML bien construido

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Además, en el XML bien construido:

- 1 Toda etiqueta abierta debe cerrarse.
- 2 Se distingue nombres en minúscula y mayúscula.
- 3 Se debe respetar el anidamiento.
- 4 Hay un sólo elemento raíz.



# XML válido

XML permite especificar el contenido de cada elemento mediante un **esquema de marcado**

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML válido

XML permite especificar el contenido de cada elemento mediante un **esquema de marcado**

Podemos determinar qué elementos pueden aparecer dentro de otro. También el orden o si son opcionales.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML válido

Que un documento sea XML bien construido no implica que sea **válido** sintácticamente. Esto dependerá de las normas sintácticas (*esquema*) a las que se adhiera.

Estándares y  
metainformación

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML válido

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Que un documento sea XML bien construido no implica que sea **válido** sintácticamente. Esto dependerá de las normas sintácticas (*esquema*) a las que se adhiera.

No deben confundirse el *análisis sintáctico (validación)* y el **procesamiento** de un documento, pues pueden realizarse independientemente.

# XML válido y bien construido

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE bibfile SYSTEM "bibfile.dtd" >
<bibfile>
  <article key="0001">
    <author>
      T. Berners-Lee, J. Hendler & O. Lassila
    </author>
    <title>The semantic web</title>
    <journal>Scientific American</journal>
    <year>2001</year>
  </article>
</bibfile>
```

Ejercicio: señala texto, etiquetas, referencias, instrucciones y declaraciones en el documento.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Instrucciones para el procesador

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Las **instrucciones para el procesador** pasan información a las aplicaciones:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<?WP (NEW PAGE)?>
```

# Etiquetas: elementos

Un par de etiquetas define un **elemento** cuyo contenido puede ser

- texto:

```
<par>Contenido de un párrafo</par>
```

- elementos:

```
<chapter>  
    <section> blablabla... </section>  
    <section> blablabla... </section>  
</chapter>
```

- mixto:

```
<par> Texto con <footnote>pie</footnote> </par>
```

- vacío: `<p></p>` (se recomienda escribir `<p/>`).

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Etiquetas: estructura lógica

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

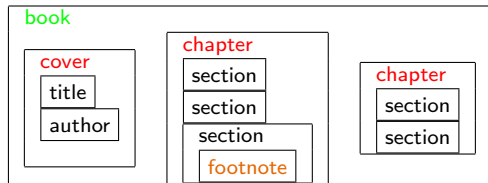
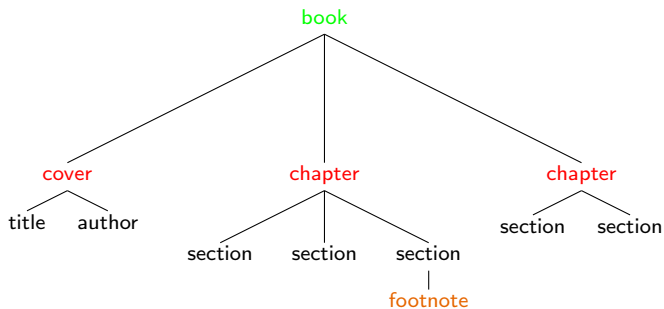
SAX Y DOM  
libxml2

Las etiquetas definen los elementos que, a su vez, definen la **estructura lógica** del documento.

Los elementos deben estar completamente contenidos en otros (prohibido `<tag1> <tag2> </tag1> </tag2>`).



# Etiquetas: estructura lógica



Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

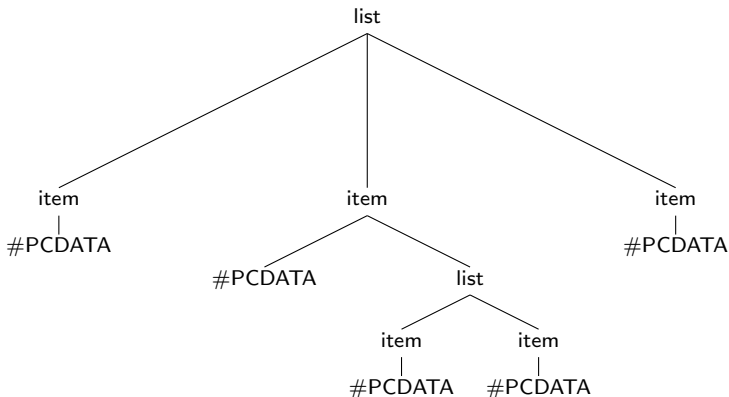
XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Etiquetas: estructura lógica

La estructura jerárquica permite recursividad:



Estándares y  
metainformación

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Etiquetas: estructura lógica

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

**Especificación**

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

¿Puede un elemento vacío tener espacios en blanco?

# Etiquetas: atributos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- Es posible guardar metainformación en forma de **atributos**:  

```
<book language="spanish" century="19" > ... </book>
```
- Siempre entrecomillados, pues pueden contener espacios en blanco.
- Se puede especificar (débilmente) el tipo de los atributos.

# Etiquetas: elementos y atributos

Si bien los atributos no aumentan el tamaño de la estructura, ¿cuándo es preferible usar elementos en vez de atributos?

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

**Especificación**

Estructura lógica  
Estructura física  
Hojas de estilo

**Sintaxis**

XMLSchema  
RELAX NG

**Semántica**

**Xtandards**

XPath  
XQuery  
XQuery  
XLink

**Procesamiento**

SAX Y DOM  
libxml2

# Etiquetas: elementos y atributos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Si bien los atributos no aumentan el tamaño de la estructura, ¿cuándo es preferible usar elementos en vez de atributos?

- Si las cadenas de texto van a ser muy largas.

# Etiquetas: elementos y atributos

Estándares y  
metainformación

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Si bien los atributos no aumentan el tamaño de la estructura, ¿cuándo es preferible usar elementos en vez de atributos?

- Si las cadenas de texto van a ser muy largas.
- Si pueden tener estructura.

# Etiquetas: elementos y atributos

Estándares y  
metainformación

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Si bien los atributos no aumentan el tamaño de la estructura, ¿cuándo es preferible usar elementos en vez de atributos?

- Si las cadenas de texto van a ser muy largas.
- Si pueden tener estructura.
- Si puede haber más de un atributo del mismo tipo (p.ej., varios autores).

Por contra permiten incluir información volátil sin tener que rediseñar la DTD.



# Etiquetas: tipos de atributos

- **Reservados:** xml:lang, xml:space,...
- **CDATA:** character data (cadenas de caracteres).
- **NMTOKEN, NMTOKENS:** como el anterior, pero restringido a nombres válidos.
- **ID, IDREF, IDREFS:** hipervínculos.
- **ENTITY, ENTITIES:** referencia a una entidad (p.ej., dibujo).
- **NOTATION:** el elemento contiene otro tipo de datos (p.ej., GIF).

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Etiquetas: tipos de atributos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

¿Se admiten referencias hacia adelante y hacia atrás?

# Comentarios y declaraciones

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- Comentarios:  
`<!-- Esto es un comentario -->`
- Declaraciones (instrucciones para el analizador):
  - [CDATA], [INCLUDE], [IGNORE]
  - DOCTYPE, ELEMENT, ATTLIST
  - ENTITY, NOTATION

# Declaraciones

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- CDATA marca zonas que no son XML:  
`<![CDATA[La flecha no es marcado -->]]>`
- IGNORE, INCLUDE marcan partes condicionales del documento: `<![IGNORE[ ...]]>`

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- DOCTYPE, ELEMENT, ATTLIST: usadas en la **Document Type Definition** o DTD para definir la **estructura lógica**.

# Declaraciones

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

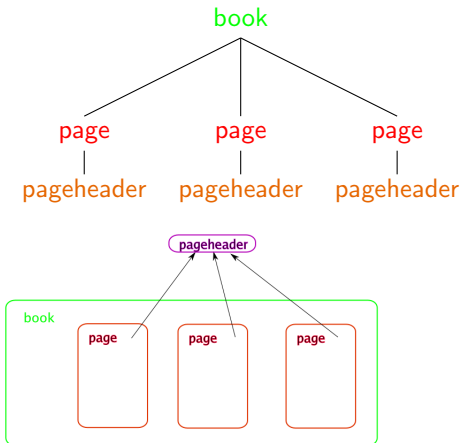
Procesamiento

SAX Y DOM  
libxml2

- DOCTYPE, ELEMENT, ATTLIST: usadas en la **Document Type Definition** o DTD para definir la **estructura lógica**.
- ENTITY, NOTATION: entidades para hacer referencia a ellas. Definen la **estructura física**.

# Estructura lógica y física

Los elementos definen la estructura lógica, las **entidades** la estructura física.



Estándares y  
metainformación

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

**Estructura lógica**  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

**Document Type Definition (DTD)**: es el conjunto de reglas gramaticales que debe obedecer el documento para ser XML válido.



Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis  
XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

**Document Type Definition (DTD)**: es el conjunto de reglas gramaticales que debe obedecer el documento para ser XML válido.

La **DTD** especifica:

- 1 qué etiquetas se pueden utilizar y
- 2 dónde pueden aparecer dichas etiquetas.

# Documentos válidos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

**Estructura lógica**  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Para que un documento sea **válido** sintácticamente debe:

- estar bien construido;

# Documentos válidos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Para que un documento sea **válido** sintácticamente debe:

- estar bien construido;
- incluir una DTD;

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Para que un documento sea **válido** sintácticamente debe:

- estar bien construido;
- incluir una DTD;
- la etiqueta del elemento raíz debe ser la definida por DOCTYPE;

# Documentos válidos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Para que un documento sea **válido** sintácticamente debe:

- estar bien construido;
- incluir una DTD;
- la etiqueta del elemento raíz debe ser la definida por DOCTYPE;
- ajustarse en su contenido a lo permitido por la DTD.

# Documentos válidos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Para que un documento sea **válido** sintácticamente debe:

- estar bien construido;
- incluir una DTD;
- la etiqueta del elemento raíz debe ser la definida por DOCTYPE;
- ajustarse en su contenido a lo permitido por la DTD.

# DTD y estructura lógica

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis  
XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

## La DTD puede ser

- **externa:**

```
<!DOCTYPE book SYSTEM "/home/guest/book.dtd">
```

- **interna:**

```
<!DOCTYPE book [<!ELEMENT book (cover,chapter+)> ...]>
```

- **con parte interna y externa:**

```
<!DOCTYPE book SYSTEM "external.dtd" [  
    declaraciones internas]>
```

Las declaraciones internas reescriben las externas

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

**Estructura lógica**

Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2



## Las declaraciones internas reescriben las externas

El uso de DTD externas:

- Reduce el espacio que ocupan los documentos.
- Normaliza los documentos.
- Evita errores.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# DTD: ejemplo

Estándares y  
metainformación

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica

Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
<!ELEMENT bibfile (article)* >
<!ELEMENT article (author, title, journal, year, key?)
<!ATTLIST article
key ID #REQUIRED >

<!ELEMENT and EMPTY>
<!ELEMENT author (#PCDATA | and)* >
<!ELEMENT title (#PCDATA) >
<!ELEMENT journal (#PCDATA) >
<!ELEMENT year (#PCDATA) >
<!ELEMENT key (#PCDATA) >
```

# DTD: modelos de contenido

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

---

---

, seguido de

---

| o

---

? opcional

---

+ una o más veces

---

\* tantos como se quiera (incluso ninguno)

---

---

# Ejercicio

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

**Estructura lógica**  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

¿Importa en qué orden se definen los elementos?

# Ejercicio

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Construye (parte de) una DTD para guardar bibliografía.  
Puedes consultar <http://bibtexml.sf.net>. Valida un fichero XML con la DTD.

Escribe las ventajas e inconvenientes de usar un formato XML válido en vez de bibtex.

# DTD y modelos de contenido

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis  
XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Los **modelos de contenido** pueden ser:

- de elementos: `<!ELEMENT book (cover, chapter+)>`
- de texto: `<!ELEMENT title (#PCDATA)>`
- vacíos: `<!ELEMENT p EMPTY>`
- mixtos: `<!ELEMENT section (#PCDATA| footnote)* >`

# DTD y modelos de contenido

Los modelos de contenido mixto tienen fuertes restricciones:

```
"(#PCDATA ("|" elementName)* ")*"
```

Por ejemplo:

```
(#PCDATA | element1 | element2 )*
```

Además:

En el estándar SGML, los modelos de contenido deben ser **no ambiguos**.

```
<!ELEMENT list1 (item,item?) > <!--unambiguous-->  
<!ELEMENT list2 (item?,item) > <!--ambiguous-->
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación  
Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis  
XMLSchema  
RELAX NG

Semántica

Xtandards  
XPath  
XQuery  
XQuery  
XLink

Procesamiento  
SAX Y DOM  
libxml2

# DTD y atributos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

**Estructura lógica**  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

ATTLIST permite restringir el tipo de atributos que puede tener un elemento. Por ejemplo,  
`<!ATTLIST book lang CDATA>`



# Tipos de atributos

**#REQUIRED** : obligatorio.

**#IMPLIED** : opcional.

**# FIXED** : fijo, constante.

**enumerado** : opciones limitadas.

**default** : valor por defecto.

```
<!ATTLIST book lang CDATA #REQUIRED>
```

```
<!ATTLIST book lang (eng|sp) #REQUIRED>
```

```
<!ATTLIST book lang #FIXED "eng">
```

```
<!ATTLIST book lang (eng|sp) "eng">
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Estructura física: referencias

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
**Estructura física**  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- Algunas entidades están predefinidas: `&lt;`; denota “<”.
- Otras se definen mediante una declaración **dentro de la DTD**:  
`<!ENTITY xml "eXtensible Markup Language">`  
Para usarlas después: El `&xml;` es divertido

¿Se puede usar una entidad antes de definirla?

# Estructura física: referencias

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
**Estructura física**  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Las entidades permiten:

- Evitar repeticiones (y errores) de información.
- Subdividir la información en fragmentos.
- Utilizar distintos formatos.

# Referencias y entidades

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
**Estructura física**  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Una referencia puede referirse a una entidad:

- interna (si está antes en el mismo documento),
- externas (si está en otro fichero).

En caso de que la entidad se encuentre definida más de una vez, sólo se considerará la primera.

# Referencias y entidades

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
**Estructura física**  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Las entidades se pueden ser:

- Un carácter o cadena de texto (predefinida o no).
- Una parte de la DTD (entidad parámetro).
- Externa con contenido XML.
- Externa con contenido binario.

# Entidades predefinidas y de carácter

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
**Estructura física**  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- `&lt;`; se refiere a “<”.
- `&gt;`; se refiere a “>”.
- `&amp;`; se refiere a “&”.
- `&quot;`; se refiere a las comillas.
- `&#169;`; se refiere a “©”.

# Entidades texto y parámetro

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- **Texto:** `<!ENTITY xml "eXtensible Markup Language">`  
El `&xml;` es divertido.
- **Parámetro** (sólo se usan en la DTD):  
`<!ENTITY %par (line+,lb)>`  
`<!ELEMENT page (%par;*) >`  
`<!ELEMENT footnote (%par;*)`

# Entidades externas

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Las **entidades externas** pueden ser XML o binarias. Si son XML (y locales), se usan como sigue:

```
<!-- primero de declara dentro de la DTD -->  
<!ENTITY signat SYSTEM "/home/guest/signature.xml">
```

```
<!-- luego se usa dentro del texto -->  
Atentamente,  
&signat;
```



# Entidades binarias

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Algunos autores creen que las entidades binarias no deberían usarse en XML.

- Es preciso determinar la aplicación que las va a tratar mediante NOTATION:  
`<!NOTATION GIF SYSTEM "/usr/bin/xv">`
- Se define una entidad de tipo NDATA (Notational Data):  
`<!ENTITY LogoUA SYSTEM "/pics/logoua.gif" NDATA GIF>`
- Y se usa en elementos vacíos que admitan atributos de tipo entidad.  
El DLSI pertenece a la `<pic name="LogoUA"/>`

# Uso de entidades públicas

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
**Estructura física**  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

XML permite el uso de entidades y de DTD públicas mediante el uso de la palabra PUBLIC seguida de un **Formal Public Identifier** (FPI) y un **URI**:

```
<!DOCTYPE HTML PUBLIC
    "-//W3C//DTD HTML 4.0 Transitional//EN"
    "http://www.w3.org/TR/REC-html40/loose.dtd">
```

# Entidades externas: ejemplo de uso

```
<!ENTITY % ISOLat1 PUBLIC
    "ISO 8879:1986//ENTITIES Added Latin 1/EN//XML"
    "/home/entities/ISOLat1.ent">
    % ISOLat1;
```

```
<!-- Includes a list of definitions:
    <!ENTITY aacute "&#x00E1;">
    <!-- SMALL LETTER A WITH ACUTE -->
-->
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
**Estructura física**  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Formal Public Identifiers

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
**Estructura física**  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

ISO	La DTD es un estándar aprobado ISO.
+	La DTD es un estándar aprobado no ISO.
-	La DTD es un estándar no aprobado.

# Espacios de nombres

Los espacios de nombres permiten que un documento tenga partes que siguen una DTD y partes que sigan otra.

```
<!DOCTYPE bibtex:file PUBLIC
    "-//BibTeXXML//DTD XML for BibTeX v1.0//EN"
    "bibteXML.dtd" >
  <bibtex:file xmlns:bibtex="http://bibtexml.sf.net/">
    ...
  </bibtex:file>
```

Por ejemplo, si queremos escribir un documento en XHTML con partes en MusiXML; o XSLT que genera XHTML.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
**Estructura física**  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Espacios de nombres

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
**Estructura física**  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

El uso de espacios de nombres genera XML bien construido y puede ser procesado por un procesador que no entienda espacios de nombres.

Por contra, el uso de distintos espacios de nombres dificulta la validación de los documentos.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- CSS = Cascading Style Sheets. Permite definir el aspecto con el que se presentará del documento.
- XSL = XML Stylesheet Language. Permite definir formatos. Es como el CSS de XML, aunque más potente.
- XSLT = XSL Transformation. Permite transformar XML en XML (p.ej. XHTML).

# Uso de CSS

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis  
XMLSchema  
RELAX NG

Semántica

Xtandards  
XPath  
XQuery  
XQuery  
XLink

Procesamiento  
SAX Y DOM  
libxml2

**bibtex.xml:**

```
<?xml version="1.0" encoding="utf-8?>  
<?xml-stylesheet href="bibtex.css" type="text/css"?>
```

**bibtex.css:**

```
* { font-family: verdana; font-size:10pt; display:block; }  
journal:before { content:"Reference: "; }  
proceedings editor:before { content:"("; }  
proceedings editor:after { content:" editor)"; }
```



# Uso de XSLT

**bibtex.xml:**

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="bibtex.xsl" type="text/xsl"?>
```

**bibtex.xsl:**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
    xmlns="http://www.w3.org/1999/xhtml1"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" media-type="text/html"/>
  <xsl:template match="/">
    <html>
      <xsl:apply-templates>
    </html>
  </xsl:template>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# Uso de XSLT con CSS

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
<?xml version="1.0"?>
<xsl:stylesheet version="2.0"
xmlns:bibtex="http://bibtexml.sf.net/"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">
    <xsl:processing-instruction name="xml-stylesheet"
      href="bibtex.css" type="text/css" />
  </xsl:processing-instruction>
  <xsl:apply-templates />
</xsl:template>
```

# Un ejemplo de XSLT

```
<xsl:template match = "/">
  <html>
    <head>References</head>
    <body><xsl:apply-templates/></body>
  </html>
</xsl:template>

<xsl:template match="bibtex:article">
  <xsl:if test="(number(bibtex:year/text())>1996)">
    <xsl:apply-templates select="*[name()!='bibtex:year']">
      <xsl:apply-templates select="bibtex:year"/>
    </xsl:if>
  </xsl:template>

<xsl:template match="@*|node()">
  <xsl:copy>
    <xsl:apply-templates select="@*|node()"/>
  </xsl:copy>
</xsl:template>
```

# Un ejemplo de XSLT output

```
<html xmlns="http://www.w3.org/1999/xhtml1">
  <head> References </head>
  <body>
    <bibtex:entry>
      <bibtex:title>The semantic web</bibtex:article>
      <bibtex:author1>T. Berners-Lee et al.<bibtex:author>
      <bibtex:journal>Scientific American</bibtex:journal>
      <bibtex:year>2001</bibtex:year>
    </bibtex:entry>
  </body>
</html>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XSLT y XHTML

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Se puede definir el espacio de nombres `xmlns` en un documento XHTML e incluir instrucciones de XSLT. Así, al menos, el analizador nos valida todas las partes del documento del espacio `xhtml`.

# Ejercicio

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Escribe un fichero xhtml que incorpore una cabecera (esto es, otro fichero válido de xml).

# XML Schema

- Es XML ⇒
  - No necesita herramientas específicas para edición, o validación.
  - Tiene DTD (?) y Schema (!).

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML Schema

- Es XML ⇒
  - No necesita herramientas específicas para edición, o validación.
  - Tiene DTD (?) y Schema (!).
- Permite especificar tipos de datos (DTD= 10, Schema> 44).

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2



# XML Schema

- Es XML ⇒
  - No necesita herramientas específicas para edición, o validación.
  - Tiene DTD (?) y Schema (!).
- Permite especificar tipos de datos (DTD= 10, Schema> 44).
- Es abierto (permite usar elementos sin definir);

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML Schema

- Es XML ⇒
  - No necesita herramientas específicas para edición, o validación.
  - Tiene DTD (?) y Schema (!).
- Permite especificar tipos de datos (DTD= 10, Schema> 44).
- Es abierto (permite usar elementos sin definir);
- MS ha prometido seguir el estándar.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML Schema

- Es XML ⇒
  - No necesita herramientas específicas para edición, o validación.
  - Tiene DTD (?) y Schema (!).
- Permite especificar tipos de datos (DTD= 10, Schema> 44).
- Es abierto (permite usar elementos sin definir);
- MS ha prometido seguir el estándar.

Tanto DTD como XMLSchema o RELAXNG son esquemas (schemata = plural de schema)

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML Schema

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtstandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

¿Cómo se vincula un XMLSchema a un documento XML?

```
<doc xmlns="http://www.w3schools.com"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://www.w3schools.com note.xsd">
```

El espacio xsi permite usar el atributo xsi:schemaLocation en el documento XML, aunque no esté permitido por el esquema. Contiene “namespace” y “schemalocation”.

# XML Schema: contenido básico

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

El espacio de nombres del XMLSchema es usualmente xsd:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

# XML Schema: contenido básico

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Los elementos se declaran mediante `xsd:element`:

```
<xsd:element name="comment" type="xsd:string" fixed="!"/>  
<xsd:element name="book" type="BookType" default="ebook"/>
```

y los atributos mediante `xsd:attribute`

```
<xsd:attribute name="lang" type="xsd:string" use="optional"/>
```

# XML Schema: definición de tipos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

De los tipos básicos pueden derivarse nuevos tipos, simples o complejos.

Los tipos simples se definen mediante `xsd:simpleType` y los complejos con `xsd:complexType`

# XML Schema: contenido básico

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

## 1 Definición de tipos de contenido:



# XML Schema: contenido básico

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- 1 Definición de tipos de contenido:
  - simpleType: elementos que continen sólo texto;

# XML Schema: contenido básico

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- 1 Definición de tipos de contenido:
  - simpleType: elementos que continen sólo texto;
  - complexType: elementos con subelementos o atributos;

# XML Schema: contenido básico

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- 1 Definición de tipos de contenido:
  - simpleType: elementos que continen sólo texto;
  - complexType: elementos con subelementos o atributos;
- 2 Declaración de contenidos:

# XML Schema: contenido básico

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- 1 Definición de tipos de contenido:
  - simpleType: elementos que continen sólo texto;
  - complexType: elementos con subelementos o atributos;
- 2 Declaración de contenidos:
  - element, group;

# XML Schema: contenido básico

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- 1 Definición de tipos de contenido:
  - simpleType: elementos que continen sólo texto;
  - complexType: elementos con subelementos o atributos;
- 2 Declaración de contenidos:
  - element, group;
  - attribute, attributeGroup.

# XML Schema: definición de tipos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

	Primitivos	Derivados
Built-in	Si	Si
User-defined	No	Si

# XML Schema: Tipos primitivos

string, boolean, decimal, float, double  
duration, dateTime, time, date  
gYearMonth, gYear, gMonthDay, gDay, gMonth  
hexBinary, base64Binary  
anyURI  
QName  
NOTATION

g= Gregoriano

100=1.0E2, pero existe forma canónica

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XMLSchema: Tipos derivados

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

normalizedString, token, Name, NCName  
language, NMTOKEN, NMTOKENS  
ID, IDREF, IDREFS, ENTITY, ENTITIES  
integer  
nonNegativeInteger, positiveInteger, nonPositiveInteger, negativeInteger  
long, int, short  
unsignedLong, unsignedInt, unsignedShort  
byte, unsignedByte

normalizedString: sin espacios en blanco



# XML Schema: Derivación de tipos por restricción

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
<xsd:simpleType name="myInteger">  
  <xsd:restriction base="xsd:integer">  
    <xsd:minInclusive value="10000"/>  
    <xsd:maxInclusive value="99999"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

# XML Schema: Derivación de tipos por restricción

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
<xsd:simpleType name="DNI">  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="\d{8}-[A-Z]"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

# XML Schema: Derivación de tipos por restricción

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
<xsd:simpleType name="month">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="enero"/>  
    . . . .  
  </xsd:restriction>  
</xsd:simpleType>
```

# XML Schema: Derivación de tipos por listas

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
<xsd:simpleType name="sizes">  
  <xsd:list itemType="decimal"/>  
</xsd:simpleType>
```

```
<sizes> 8 10.5 12 </sizes>
```

# XML Schema: derivación de tipos complejos

```
<xsd:complexType name="BookType">
  <xsd:sequence>
    <xsd:element name="title" type="xsd:string"/>
    <xsd:element name="author" type="xsd:string"/>
    <xsd:element name="year" type="xsd:gYEAR"/>
    <xsd:element name="publisher" type="xsd:string" />
    <xsd:element ref="comment" />
  </xsd:sequence>
  <xsd:attribute name="ISBN" type="xsd:decimal"/>
</xsd:complexType>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML Schema: derivación de tipos complejos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

## Observaciones:

- En la definición anterior aparecen subelementos y atributos (ISBN).
- El atributo `ref` de `xsd:element` permite evitar redefiniciones.
- Un elemento puede tener el mismo nombre y distinto tipo en dos definiciones de `complexType` distintas (sensibilidad al contexto).

# XML Schema: derivación de tipos complejos

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Además de `xsd:sequence`, tenemos `xsd:union`.  
Para los operadores de repetición se usan dos atributos  
opcionales de `sequence` y `union`: `minOccurs` y `maxOccurs`

# Ejercicio

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Transforma el modelo de contenido  $(A|(B,C?))^*$  en una definición mediante XMLSchema.



# Ejercicio

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtstandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
<complexType name="modelo">  
  <choice minOccurs="0" maxOccurs="unbounded">  
    <element ref="a"/>  
    <sequence>  
      <element ref="b"/>  
      <element ref="c" minOccurs="0" maxOccurs="1"/>  
    </sequence>  
  </choice>  
</complexType>
```

# Transformación de DTD en XML Schema

*	minOccurs=0	maxOccurs=unbounded
+	minOccurs=1	maxOccurs=unbounded
?	minOccurs=0	maxOccurs=1
,	xsd:sequence	
	xsd:choice	
()	no hay	
X	xsd:element	
%X;	xsd:group ref="X"	

Los atributos minOccurs y maxOccurs son opcionales de group, sequence o choice.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML Schema: entidades parámetro

Los grupos de atributos permiten modularizar las definiciones de atributos y se usan como las entidades parámetro de una DTD:

```
<xsd:attributeGroup name="occurs"> ...  
</xsd:attributeGroup>  
<xsd:attributeGroup ref="occurs"/>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML Schema: entidades parámetro

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

¿Qué diferencia hay entre referenciar un elemento y un grupo en un modelo de contenido?

# XML Schema: derivación mediante unión

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

La definición del atributo occurs es la siguiente:

```
<xsd:attributeGroup name="occurs">  
  <xsd:attribute name="minOccurs" type="nonNegativeInteger  
    default="1"/>  
  <xsd:attribute name="maxOccurs">  
    . . . .  
</xsd:attribute> </xsd:attributeGroup>
```

# XML Schema: derivación mediante unión

```
<xsd:attribute name="maxOccurs">
  <xsd:simpleType>
    <xsd:union>
      <xsd:simpleType>
        <xsd:restriction base="nonNegativeInteger"/>
      </xsd:simpleType>
      <xsd:simpleType>
        <xsd:restriction base="string">
          <xsd:enumeration value="unbounded"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:union>
  </xsd:simpleType>
</xsd:attribute>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML Schema: derivación mediante extensión

```
<xsd:complexType name="Address">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="street" type="xsd:string"/>
    <xsd:element name="city" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UKAddress">
  <xsd:complexContent>
    <xsd:extension base="Address">
      <xsd:sequence>
        <xsd:element name="postcode" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# XML Schema: derivación mediante extensión

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
<xsd:element name="shipTo" type="Address">
```

```
<shipTo xsi:type="UKAddress">  
  <name>Helen Zoe</name>  
  <street>47 Eden Street</street>  
  <city>Cambridge</city>  
  <postcode>CB1 1JR</postcode>  
</shipTo>
```



# XML Schema: sensibilidad al contexto

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

**XMLSchema**  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Un buen ejemplo por **Sperberg-McQueen**.

- Es XML y se aprende rápidamente.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- Es XML y se aprende rápidamente.
- Basado en la teoría gramatical: sensible al contexto.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# RELAX NG

- Es XML y se aprende rápidamente.
- Basado en la teoría gramatical: sensible al contexto.
- Permite especificar tipos de datos (como XML Schema).

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# RELAX NG

- Es XML y se aprende rápidamente.
- Basado en la teoría gramatical: sensible al contexto.
- Permite especificar tipos de datos (como XML Schema).
- Es más potente que las DTD o XML Schema.

Estándares y  
metainformación

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# RELAX NG

- Es XML y se aprende rápidamente.
- Basado en la teoría gramatical: sensible al contexto.
- Permite especificar tipos de datos (como XML Schema).
- Es más potente que las DTD o XML Schema.
- No permite valores por defecto (salvo que se combine con DTD).

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- Es XML y se aprende rápidamente.
- Basado en la teoría gramatical: sensible al contexto.
- Permite especificar tipos de datos (como XML Schema).
- Es más potente que las DTD o XML Schema.
- No permite valores por defecto (salvo que se combine con DTD).
- Permite usar espacios de nombres.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- Es XML y se aprende rápidamente.
- Basado en la teoría gramatical: sensible al contexto.
- Permite especificar tipos de datos (como XML Schema).
- Es más potente que las DTD o XML Schema.
- No permite valores por defecto (salvo que se combine con DTD).
- Permite usar espacios de nombres.
- Permite contenido sin orden. Los modelos de contenido pueden ser ambiguos e incluir atributos.



# RELAX NG

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- Es XML y se aprende rápidamente.
- Basado en la teoría gramatical: sensible al contexto.
- Permite especificar tipos de datos (como XML Schema).
- Es más potente que las DTD o XML Schema.
- No permite valores por defecto (salvo que se combine con DTD).
- Permite usar espacios de nombres.
- Permite contenido sin orden. Los modelos de contenido pueden ser ambiguos e incluir atributos.
- No permite entidades ni NOTATION.

# RELAX NG: ejemplo

```
<grammar>
  <start>
    <element name="addressBook">
      <zeroOrMore>
        <element name="card">
          <ref name="cardContent"/>
        </element>
      </zeroOrMore>
    </element>
  </start>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# RELAX NG: ejemplo

```
<define name="cardContent">
  <element name="name">
    <text/>
  </element>
  <element name="email">
    <text/>
  </element>
</define>
</grammar>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# RELAX NG: sintaxis compacta

James Clark ha inventado una notación compacta (estilo DTD) para RelaxNG:

```
element addressBook {  
  element card {  
    (element name { text }  
     | (element givenName { text },  
        element familyName { text })),  
    element email { text },  
    element note { text }?  
  }*  
}
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# RELAX NG y lenguajes de árboles

Permite definir modelos de contenido dependientes del contexto.

```
<grammar>
  <start>
    <element name="doc">
      <ref name="parWithoutFNotes"/>
      <zeroOrMore>
        <ref name="parWithFNotes"/>
      </zeroOrMore>
    </element>
  </start>
  ...
</grammar>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# RELAX NG y lenguajes de árboles

```
<define name="parWithoutFNotes">  
  <element name="para">  
    <text/>  
  </element>  
</define>
```

```
<define name="parWithFNotes">  
  <element name="para">  
    <mixed>  
      <zeroOrMore>  
        <ref name="fnote"/>  
      </zeroOrMore>  
    </mixed>  
  </element>  
</define>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# RELAX NG y lenguajes de árboles

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

En sintaxis compacta:

```
start = element doc { parWithoutFNotes, parWithFNotes* }
```

```
parWithoutFNotes = element para { text }
```

```
parWithFNotes = element para { mixed { fnote* } }
```

# XML y relaciones semánticas

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

**Semántica**

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Un ordenador no comprende al información que procesa. Por ello, comprar una entrada para la película “Star wars” requiere un esfuerzo considerable.



# XML y relaciones semánticas

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Un ordenador no comprende al información que procesa. Por ello, comprar una entrada para la película “Star wars” requiere un esfuerzo considerable.

Hay que localizar cines, películas y precios; filtrar bases de datos ([www.imdb.com](http://www.imdb.com)), revistas, páginas personales, etc.

# XML y relaciones semánticas

El **RDF** (Resource Description Format) es un formato para la descripción de documentos de Internet.

- 1 Es una aplicación del XML (las “fichas” de Internet).
- 2 Se suele usar en las bibliotecas digitales con el vocabulario de Dublin Core (Dublin, Ohio).
- 3 Pretende añadir significado (interoperabilidad) a Internet.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

## XML y relaciones semánticas ¿Por qué no usar simplemente un vocabulario de XML?

- 1 El orden no es importante en la metainformación sobre autor, fecha de publicación etc.
- 2 Debe ser abierto para adaptarse a la metainformación disponible.

# RDF y Dublin Core

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

**Semántica**

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

RDF describe propiedades de un recurso o relaciones entre ellos. Por ejemplo,

*El tema de <http://www.cervantesvirtual.com> es la literatura hispánica*

# RDF y Dublin Core

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

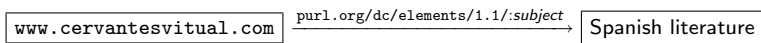
Procesamiento

SAX Y DOM  
libxml2

RDF describe propiedades de un recurso o relaciones entre ellos. Por ejemplo,

*El tema de <http://www.cervantesvital.com> es la literatura hispánica*

puede representarse como una relación



# RDF y Dublin Core

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
<rdf:RDF xmlns:rdf="http://www.w3.org/RDF/"  
          xmlns:dc=' 'http://purl.org/RDF/DC/">  
  <rdf:Description  
    rdf:about="http://www.cervantesvirtual.com">  
    <dc:subject>Literature</dc:subject>  
  </rdf:Description>  
</rdf:RDF>
```

# RDF y Dublin Core

Dublin Core (<http://dublincore.org/>) es un estándar para metadatos basado en RDF

Estándares y metainformación

Aspecto y estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# RDF y Dublin Core

Dublin Core (<http://dublincore.org/>) es un estándar para metadatos basado en RDF

title  
subject  
description  
creator  
publisher  
contributors  
date

type  
format  
identifier  
source  
language  
relation  
coverage  
rights

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2



XPath permite identificar elementos del documento XML.  
Aquí puedes encontrar algunos **ejemplos**.

```
XPathExpression xp =  
    XPathFactory.newInstance().newXPath();  
NodeList nodes =  
    (NodeList)xp.evaluate(path, doc.getDocumentElement(),  
        XPathConstants.NODESET );  
for (int k = 0; k < nodes.getLength(); ++k)  
    System.out.println( nodes.item(k) );
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

**XPath**

XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Permite seleccionar los nodos con un cierto contenido:

```
'//year[child::text()="2005"]'
```

# XInclude

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
<a>  
  <xi:include href="b.xml"  
             xmlns:xi="http://www.w3.org/2001/XInclude"/>  
</a>
```

Si b.xml es ‘‘<b/>’’, entonces `xmllint --xinclude a.xml`  
produce

```
<a>  
  <b/>  
</a>
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
**XQuery**  
XLink

Procesamiento

SAX Y DOM  
libxml2

Expresiones FLWOR (For, Let, Where, Order by, Return):

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

No es XML!

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
**XQuery**  
XLink

Procesamiento

SAX Y DOM  
libxml2

Permite bloques condicionales e inserción de etiquetas:

```
for $x in doc("books.xml")/bookstore/book
return if ($x/@category="CHILDREN")
       then <child>{data($x/title)}</child>
       else <adult>{data($x/title)}</adult>
```

# XLink y XPointer

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

XLink permite vínculos generalizados:

```
<description
  xlink:type="simple"
  xlink:href="http://book.com/HPotter#Index"
  xlink:show="new">
  ....
</description>
```

XPointer permite denotar elementos: en este caso aquél con atributo `id="Index"`.

# Herramientas: Java

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Java tiene clases para manipular y validar XML (DTD y RelaxNG).

También incluye expresiones XPath.

Escribe un programa en Java que lea el nombre de un documento XML y una expresión XPath y escriba los subdocumentos (si hay) que casan con la expresión.

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- 1 SAX (Simple API for XML) es un conjunto de procedimientos para procesar secuencial XML.
- 2 DOM (Document Object Model) es un conjunto de procedimientos para manipular árboles.



# ¿SAX o DOM?

random access time

SAX

DOM

space

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# ¿SAX o DOM?

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

- SAX es rápido para leer, pero no permite acceso aleatorio.
- DOM permite escribir, pero requiere más memoria.

El programador de aplicaciones SAX sólo debe implementar funciones del tipo:

```
public void  
endElement (String URI, String localName, String qName)  
    throws SAXException {  
    System.out.println("</" + qName + ">")  
}
```

Las no implementadas no hacen nada.

# SAX

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
private XMLReader getXMLReader() {
    XMLReader reader = null;
    try {
        reader =
            SAXParserFactory.newInstance().newSAXParser().getXMLReader();
        reader.setContentHandler( this );
    } catch (Exception x) {
        System.err.println( x.getMessage() );
    }
    return reader;
}

public String getText( String fileName ) {
    XMLReader reader = getXMLReader();
    try { reader.parse(fileName); }
```

DOM no es una estructura de datos ni un formato de almacenamiento

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

DOM no es una estructura de datos ni un formato de almacenamiento

DOM está descrito de forma “neutral” respecto a los lenguajes (IDL), pero contiene especificación para Java.

Estándares y metainformación

Aspecto y estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

DOM no es una estructura de datos ni un formato de almacenamiento

DOM está descrito de forma “neutral” respecto a los lenguajes (IDL), pero contiene especificación para Java.

Escribe un programa que escriba el árbol de un documento XML.

# DOM

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
DocumentBuilderFactory dbf =
    DocumentBuilderFactory.newInstance();
DocumentBuilder builder;
    try {
        builder = dbf.newDocumentBuilder();
        doc = builder.parse( fileName );
    }
...
    NodeList children = node.getChildNodes();
    int len = children.getLength();
    for ( int i = 0; i < len; ++i )
        printTree( children.item(i) );
```



Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

`libxml` es un conjunto de funciones de código abierto en C que implementan

- el estándar XML;
- XPath, XPointer, Xinclude;
- Relax NG, W3C Schemas.

# libxml2

```
typedef enum {
    XML_ELEMENT_NODE=          1,
    XML_ATTRIBUTE_NODE=        2,
    XML_TEXT_NODE=              3,
    XML_CDATA_SECTION_NODE=     4,
    XML_ENTITY_REF_NODE=        5,
    XML_ENTITY_NODE=            6,
    XML_PI_NODE=                7,
    XML_COMMENT_NODE=           8,
    XML_DOCUMENT_NODE=          9,
    XML_DOCUMENT_TYPE_NODE=     10,
    XML_DOCUMENT_FRAG_NODE=     11,
    XML_NOTATION_NODE=          12,
    XML_HTML_DOCUMENT_NODE=     13,
    XML_DTD_NODE=               14,
    XML_ELEMENT_DECL=           15,
    XML_ATTRIBUTE_DECL=         16,
    XML_ENTITY_DECL=            17,
    XML_NAMESPACE_DECL=         18,
    XML_XINCLUDE_START=         19,
    XML_XINCLUDE_END=           20
} xmlElementType;
```

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

# libxml2

```
struct xmlNode {
    void          *_private;      /* application data */
    xmlElementType  type;        /* type number, must be second
    const xmlChar   *name;        /* the name of the node, or NULL
    struct _xmlNode *children;    /* parent->childs link */
    struct _xmlNode *last;       /* last child link */
    struct _xmlNode *parent;     /* child->parent link */
    struct _xmlNode *next;       /* next sibling link */
    struct _xmlNode *prev;       /* previous sibling link */
    struct _xmlDoc  *doc;        /* the containing document */

    xmlChar        *content;      /* the content */
    struct _xmlAttr *properties; /* properties list */
}
```

# libxml2

Estándares y  
metainforma-  
ción

Aspecto y  
estructura  
Prospectiva

Especificación

Estructura lógica  
Estructura física  
Hojas de estilo

Sintaxis

XMLSchema  
RELAX NG

Semántica

Xtandards

XPath  
XQuery  
XQuery  
XLink

Procesamiento

SAX Y DOM  
libxml2

```
struct xmlElementContent {
    xmlElementType      type;      /* PCDATA, ELEMENT, ... */
    xmlElementContentOccur  ocur;   /* ONCE, OPT, MULT o... */
    const xmlChar        *name;     /* Element name */
    struct _xmlElementContent *c1;  /* first child */
    struct _xmlElementContent *c2;  /* second child */
    struct _xmlElementContent *parent; /* parent */
    const xmlChar        *prefix;   /* Element name */
};
```

## Estándares y metainformación

- Aspecto y estructura
- Prospectiva

## Especificación

- Estructura lógica
- Estructura física
- Hojas de estilo

## Sintaxis

- XMLSchema
- RELAX NG

## Semántica

## Xtandards

- XPath
- XQuery
- XQuery
- XLink

## Procesamiento

- SAX Y DOM
- libxml2

Información adicional:

<http://www.dlsi.ua.es/asignaturas/doctorado/mt>