



---

# Curso de Web Semántica

Maestría en Computación- Universidad del Cauca

---

## XML y RDF en Web Semántica

MSC. MIGEL ANGEL NIÑO ZAMBRANO

# Agenda

- Clasificación de los Metadatos
- Lenguaje de Marcas
- XML
  - Conceptos
  - Historia y Objetivos
  - SGML,HTML, DHTML vs. XML
  - Características del XML
  - Estructura del XML
  - Aplicaciones XML
- Ejemplos HTML, XHTML y XML (Código)
- RDF
  - Conceptos
  - Estructura del RDF
  - Microformatos
  - Diferencias entre XML y RDF
- Ejemplos RDF (Código)

# Los Metadatos

Tipo	Objetivo	Ejemplos
<b>Descriptivos</b>	<i>Describen e identifican recursos de información. Permite a los usuarios la búsqueda y recuperación de la información.</i>	Dublin Core o Etiquetas META de HTML
<b>Estructurales</b>	<i>Facilitan la navegación y la presentación de los recursos. Proporcionan información sobre la estructura interna de los documentos, así como la relación entre ellos.</i>	XML y RDF o SGML
<b>Administrativos</b>	<i>Facilitan la gestión de conjuntos de recursos. Incluye la gestión de derechos y sobre control de acceso y uso.</i>	MOA2

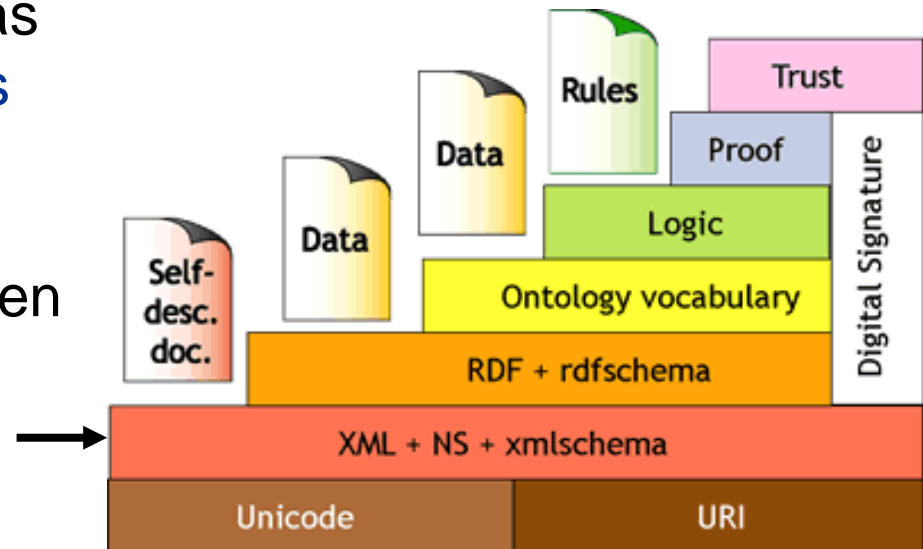
Fuente: Metadatos y Documentos XML/RDF para Recuperación

# Lenguaje de Marcas

- 1960 – IBM propone el GML (Generalized markup Language).
- SGML ( ISO 8879), Standart Generalized Markup Language. Ej. HTML y RTF.
- Difieren de los Lenguajes de programación:
  - Maneja un Editor de texto.
  - Etiquetas de abrir y cerrar. La más usada (<>).
- Usos de los lenguajes de marcas:
  - Describir contenidos (Bases de Datos).
  - Definir el formato de los datos (Procesadores de texto).
  - Realizar las dos funciones anteriores al tiempo (HTML).

# ¿QUE ES XML?

- XML, es el estandar de **E**xtensible **M**arkup Language. XML no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes.
- XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.



# HISTORIA Y OBJETIVOS

- XML fue creado al amparo del Word Wide Web Consortium (W3C).
- Su desarrollo se comenzó en 1996 y la primera versión salió a la luz el 10 de febrero de 1998.
- Principales Objetivos:
  - ❑ XML debe ser directamente utilizable sobre Internet.
  - ❑ Soportar una amplia variedad de aplicaciones
  - ❑ Compatible con SGML.
  - ❑ De fácil la escritura de programas que procesen documentos XML.
  - ❑ Los documentos XML deben ser legibles por humanos y razonablemente claros.

# HTML VS. XML

## ■ HTML

1. Permite Insertar menús, tablas, imágenes o bases de datos en los documentos.
2. Acceder a información de diferentes fuentes.
3. Aplicaciones a la medida (poca interoperabilidad).
4. Es un lenguaje de Marcas.
5. Carece de elementos dinámicos o lógica de ejecución, esta lógica en un navegador es llevada acabo mediante un "Scripting Language".
6. Requiere DTD.

## ■ XML

1. Estructura la información para uso del computador.
2. Accede a la información de acuerdo a una estructura formal, validándola.
3. Permite interoperabilidad sintáctica.
4. NO es HTML extendido.
5. Carece de elementos dinámicos o lógica de ejecución, esta lógica en un navegador es llevada acabo mediante un "Scripting Language".
6. No Requiere DTD.

# Tabla Comparativa HTML, XML y SGML

	HTML/DHTML	XML	SGML
<b>Gramática</b>	Fija y no ampliable	Extensible	Extensible
<b>Estructura</b>	Monolítica	Jerárquica	Jerárquica
<b>Nº de marcas</b>	Fijas	Sin límite	Sin límite
<b>Complejidad</b>	Baja	Mediana	Alta
<b>Diseño de páginas</b>	Fijado por tags. Etiquetas con atributos CSS en DHTML	CSS o XSL	DSSSL
<b>Enlaces</b>	Simples enlaces	Poderosos enlaces (XLL)	HyTime
<b>Exportabilidad (formatos/aplicaciones)</b>	No	Sí	Sí
<b>Validación</b>	Sin validación	Pueden validarse	Obligatorio DTD
<b>Búsquedas</b>	Simple y a veces resuelta por <i>scripts</i> o CGI	Potente búsqueda. Con capacidad para personalizarla	Son posibles potentes búsquedas.
<b>Indización/Catalogación de páginas web</b>	Sólo lo permite los atributos de la etiqueta <META>, e implementaciones como DC.	Una descripción abierta y personalizable con el RDF.	Algún proyecto como TEI, DLI, etc.



# Características del XML

- **Arquitectura abierta y extensible.** No se necesita versiones para que puedan funcionar en futuros navegadores.
- Mayor **consistencia, homogeneidad y amplitud** de los identificadores descriptivos del documento con XML (los **RDF** Resource Description Framework), en comparación a los atributos de la etiqueta <META> del HTML.
- **Integración de los datos** de las fuentes mas dispares.
- **Gestión y manipulación** de los datos desde el propio **cliente Web**.
- Los motores de **búsqueda** devolverán respuestas más adecuadas y **precisas**.

# Características del XML

- Se desarrollarán de manera extensible las **búsquedas personalizables** y subjetivas para **robots** y **agentes inteligentes**.
- Se permitirá un **comportamiento más estable** y actualizable de las aplicaciones Web.
- El concepto de "**hipertexto**" se desarrollará ampliamente (permitirá **denominación independiente de la ubicación**, enlaces **bidireccionales**, enlaces que pueden especificarse y gestionarse desde fuera del **documento**, hiperenlaces **múltiples**, enlaces agrupados, atributos para los enlaces, etc. Creado a través del Lenguaje de enlaces extensible (XLL).
- **Exportabilidad** a otros formatos de publicación (papel, web, cd-rom, etc.).

# Estructura del XML

- **DTD** (*Document Type Definition*):
  - Una definición formal de un **tipo de documento**
  - Especifica la **estructura lógica** de cada documento.
  - Define tanto los **elementos** de una página como sus **atributos, estructura y anidamiento**.
  - El DTD del XML es opcional. En tareas sencillas no es necesario construir una DTD, entonces se trataría de un documento "**bien formado**" (*well-formed*) y si lleva DTD será un documento "**validado**" (*valid*).

# Ejemplo de un DTD básico

ejemplo.dtd

```
<!ELEMENT lista_de_personas (persona*)>
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT fechanacimiento (#PCDATA) >
<!ELEMENT sexo (#PCDATA) >
<!ELEMENT numeroseguridadsocial (#PCDATA)>
```

1. **<lista\_de\_personas>** es un nombre de elemento válido. El \* indica que puede haber 0 o más elementos de persona.
2. **<persona>** es un nombre de elemento válido. Éste contiene obligatoriamente el elemento nombre mientras que el resto son opcionales. Y lo son porque nos lo indica el símbolo "?".
3. **<nombre>** es un nombre de elemento válido. Contiene caracteres.
4. **<fechanacimiento>** es un nombre de elemento válido.
5. **<sexo>** es un nombre de elemento válido. Contiene caracteres.
6. **<numeroseguridadsocial>** es un nombre de elemento válido.

# Ejemplo de documento XML basado en el DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE lista_de_personas SYSTEM "ejemplo.dtd">
<lista_de_personas>
  <persona>
    <nombre>Miguel Angel Niño</nombre>
    <fechanacimiento>14/01/1973</fechanacimiento>
    <sexo>masculino</sexo>
    <numeroseguridadsocial>91</numeroseguridadsocial>
  </persona>
</lista_de_personas>
```

# Estructura de XML

- **XSL** (*eXtensible Stylesheet Language*):
  - Define o implementa el **lenguaje de estilo** de los documentos escritos para XML.
  - Permite **modificar el aspecto de un documento**. Se puede lograr múltiple columnas, texto girado, orden de visualización de los datos de una tabla, múltiples tipos de letra con amplia variedad en los tamaños.
  - Este estándar está basado en el lenguaje de semántica y especificación de estilo de documento (**DSSSL**, *Document Style Semantics and Specification Language*, ISO/IEC 10179) y, por otro lado, se considera más potente que las hojas de estilo en cascada (CSS, *Cascading Style Sheets*), usado en un principio con el lenguaje DHTML.

# Estructura de XML

- **XLL** (*eXtensible Linking Language*):
  - Define el **modo de enlace entre diferentes enlaces**. Se considera que es un subconjunto de HyTime (*Hipermedia/Timed-based structuring Language* o Lenguaje de estructuración hipermedia/basado en el tiempo, ISO 10744).
  - Este lenguaje de enlaces extensible tiene dos importantes componentes: **Xlink y el Xpointer**.
- **XUA** (*XML User Agent*): **Estandarización de navegadores XML**. Todavía está en proceso de creación de borradores de trabajo. Se aplicará a los navegadores para que compartan todas las especificaciones XML.

# Ejemplos (1)

- Ejemplo 1: Creación de un documento HTML.
- Ejemplo 2: Uso de los DTD.
- Ejemplo 3: Validación de HTML:
  - HTML-Tidy : Software Open-Source utilizado para la validación de documentos HTML :  
<http://tidy.sourceforge.net/> (web anterior)  
<http://www.html-tidy.org/> (web actual)
  - Servicio de Validación en-línea : El W3C ofrece una herramienta de validación en-línea para HTML/XHTML:  
<http://validator.w3.org/>
  - Inspección Manual : Es posible consultar directamente los DTD definidos en la especificación HTML :  
<http://www.w3.org/TR/html4/>, en español:  
<http://html.conclase.net/w3c/html401-es/cover.html>.



# Ejemplos (2)

- Ejemplo 4: Cascading Style Sheets
  - Estilos a cada Tag y en línea
  - Estilos desde un archivo CSS
  - Ejemplo básico de DHTML
- Ejemplo 5: XML
  - Diferencia entre HTML, XHTML y XML
  - Usando DOM o XSL

## Ejemplos (2)

- Ejemplo 6: Consumo de XML
  - Crear un DTD Personalizado (DTDDContactos.dtd)
  - Crear una instancia (ListaContactosMiguel.xml)
  - Crear Pagina Web que utilice el XML (ListaContactos.aspx).
  - Mostrar la Creación de un XMLS a partir del XML construido (notar diferencias).

# ¿Qué ES RDF?

- El Marco de Descripción de Recursos es un **framework** para **metadatos** en la World Wide Web (WWW), desarrollado por W3C y basado en XML.
- **Reglas del RDF:**
  - Un **recurso** es cualquier cosa que puede tener un URI, esto incluye todas las páginas Web, todos los elementos individuales de cada documento XML y mucho más.
  - Una **propiedad** es un recurso que tienen un nombre y que puede usarse como una propiedad, por ejemplo autor o título.
  - Una **sentencia** consiste en la combinación de un **recurso**, una **propiedad** y un **valor**. Estas partes son conocidas como el **sujeto**, **predicado** y el **objeto** de la sentencia.
- **Ejemplo:** Una sentencia es: "El **autor** de <http://metadatos-xml-rdf.awardspace.com/rdf.html> es **Julio César Ayllón Bonet**". Este último puede ser un URI también.

# Estructura de RDF

- **Independencia** – (cada propiedad puede ser inventada).
- **Intercambio** – (Escritas en XML).
- **Escalabilidad** – (Recurso, propiedad, valor).
- **Las propiedades son recursos** – (Propiedades -> propiedades).
- **Los valores pueden ser recursos** – ("*home*" que apunte al home del sitio)
- **Las sentencias pueden ser recursos** - Esto significa que queremos, dada una sentencia como "*El tema de esta pagina es monos*" poder preguntar "*¿Quien lo dice?*", "*¿Cuando?*". Una forma útil de hacer esto es mediante metadatos y por ello las sentencias deben poder tener sus propias propiedades.

# Microformatos

- Los microformatos son **porciones de código XHTML ó HTML** estándar cuyo objetivo es insertar contenido semántico aprovechando las características de los atributos "**id**" ó "**class**" usada por algunas etiquetas de HTML.
- El objetivo principal de los microformatos, es que los metadatos sean útiles a las personas en primer lugar, y después a los agentes de usuarios (como por ejemplo los buscadores).
- Ejemplos de uso de microformatos:  
<http://www.webposible.com/microformatos-dublincore/ejemplos-microformatos-dublincore.html>.

# Diferencias entre XML y RDF

- Mientras que ***XML*** es un lenguaje para modelar datos, ***RDF*** es un lenguaje para especificar metadatos.
- ***XML*** falla en la escalabilidad de los datos puesto que el orden de los elementos es antinatural y su mantenimiento es muy difícil y costoso, por el contrario, ***RDF*** permite la interoperabilidad entre aplicaciones que intercambian información comprensible por la página Web, para proporcionar una infraestructura que soporte actividades de metadatos.

## Ejemplos (3)

- Ejemplo 7: Creación de un RDF Dublin Core directamente (RDFEjemplo1.xml, HTMLIncluyeDC.htm).
- Ejemplo 8: Uso de Microformatos DC y CSS (StyleSheetAutor.css, HTMLIncluyeDC.htm)
- Ejemplo 9: Uso de XSLT en un RDF para presentar datos semánticos (RDFWikipedia.rdf, XSLTWikipedia.xslt, RDFWikipedia.xml).
- Ejemplo 10: Creación de un RDF Dublin Core con un generador: (Proyecto CursoWebSemantica)  
<http://www.webposible.com/utilidades/dublincore-metadata-gen/>.
- Ejemplo 11: Indexar página en el Buscador:  
<http://www.taringa.net/posts/ebooks-tutoriales/1889195/Como-Indexar-Tu-Web.html>.

# Ejemplos (4)

- Trabajo de Programación XML y Bases de Datos (XMLDataSet.aspx)
  - ❑ Leer datos XML y presentarlos en un GridView.
  - ❑ Guardar Datos de Una Tabla / Vista de base de Datos en un XML.
  - ❑ Guardar varias tablas en un archivo XML sin anidar y anidadas.
- Integrar y Sincronizar XMLDataDocument y Dataset (DataSetXMLDataDoc.aspx)
  - ❑ Cargar y guardar un XMLDataDocument desde una tabla de la BD.
  - ❑ Cargar y XMLDataDocument y presentarlo en un GridView.
  - ❑ Aplicar una Transformación XSLT al XMLDataDocument.
  - ❑ Uso del Control XML del Servidor



---

# Bibliografía

- Introducción a los Microformatos de Dublin Core.

<http://www.webposible.com/microformatos-dublincore/introduccion-microformatos.html>,  
consultado 23/07/2009.

---