

# *Introducción a XML*

M. Carmen Fernández Panadero  
<mcfp@it.uc3m.es>

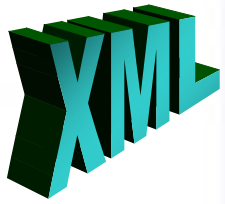




# Objetivos

- Familiarizarse con conceptos y herramientas básicas de XML
- Conocer la estructura de un documento XML
- Ser capaz de diseñar y comprender un DTD
- Ser capaz de construir un documento XML
- Ser capaz de transformar y dar formato a documentos XML utilizando hojas de estilo XSLT





# Índice

- ¿Qué es XML?
  - Diferencias con HTML y SGML
  - Definición
- ¿Cómo es XML?
  - Representación
  - Tipos de documentos
  - DTDs
- Hojas de estilo (XSLT)
- Enlaces XLL (XLink y XPath)
- Procesamiento (DOM y SAX)
- Conclusiones





# Qué es XML:

Un poco de historia

## **Problemas de HTML**

- Extensibilidad
- Estructura lógica
- Intercambio de datos
- Reutilización de datos
- Gestión de enlaces rotos.
- Etc

## **Beneficios de HTML**

- Simple
- No requiere de herramientas especiales
- Extendido
- Papel importante en crecimiento de internet.

## **Problemas de SGML**

- Complejidad

## **Beneficios de SGML**

- Reutilización de los datos
- Longevidad de la información
- Integridad y mayor control sobre los datos.
- Permite compartir información
- Portable
- Flexible

**XML = +80% -20%**





# Qué es XML

## Valor añadido

- Permite almacenar **cualquier tipo** de información estruct.
- **Comprensible** por máquinas y por humanos.
- **Independiente** de:
  - Lenguajes de programación o
  - Plataformas software.
- Permite **validar** los documentos
- **Distribución**: documento, sw que lo valida, hoja de estilo y resultado final pueden estar en distintas máquinas
- Herramientas **gratuitas** (validación, transformación, etc.)
- Lenguajes **específicos** para diferentes industrias.
- **Estandar abierto y ampliamente difundido.**





# Qué es XML:

*(EXtended Markup Language)*

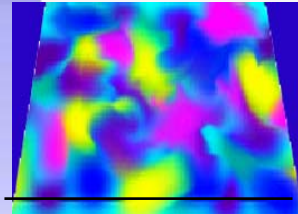
- **Lenguaje de definición de lenguajes de marcado**
  - **Metalinguaje**
  - Define lenguajes de marcado por medio de DTD's (Document Type Definition)
  - Estandarizado por el w3c (Recommendation 10-Feb- 1998)
  - No es una aplicación de SGML sino una versión reducida (+80% beneficios -20% complejidad)
- **Conjunto de especificaciones**
  - **XML** (Estructura de los datos)
  - **XSL = XSLT (utiliza XPath) + XSL-FO's** : Hojas de estilo
  - **XLL = XLink + XPointer (utiliza Xpath)** : Hiperenlaces



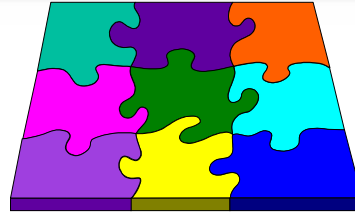


# Qué es XML:

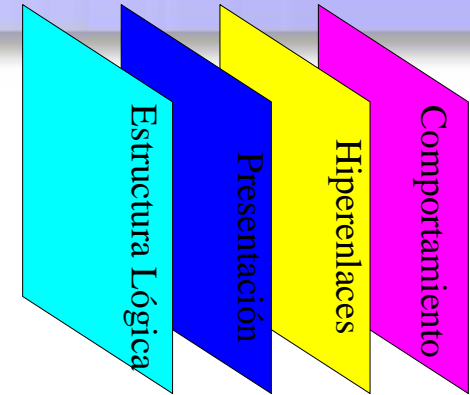
## ¿Qué es un documento?



Marcado no declarativo



Marcado declarativo



XML XSL XLL Java JavaScript

- **XML: Intercambio** de Información
- **XML+XSL:** Formato de *publicación* indep
- **XML+XSL+XLL: Navegación**
- **XML+XSL+XLL+ script:** DXML (*aplicación*)





# Cómo es XML: (XML vs HTML)

```

<html>
<dl>
  <dt>Hot Cop
  <dd>by Jacques Morali,
      Henry Beolo, and Victor Wills
  <ul>
    <li>producer: Jacques Morali </li>
    <li>publisher: PolyGramRecords</li>
    <li>length: 6:20</li>
    <li>written: 1978 </li>
    <li>artist: Village People</li>
  </ul>
</dl>
</html>

```

```

<?xml version='1.0'?>
<song>
  <title>Hot Cop</title>
  <composer> Jacques Morali</composer>
  <composer> Henry Beolo </composer>
  <composer> Victor Wills </composer>
  <producer> Jacques Morali </producer>
  <publisher> PolyGram </publisher>
  <length> 6:20 </length>
  <year> 1978 </year>
  <artist> Village People </artist>
</song>

```

Hot Cop  
 by Jacques Morali, Henry Beolo, and Victor Wills

- producer: Jacques Morali
- publisher: PolyGramRecords
- length: 6:20
- written: 1978
- artist: Village People

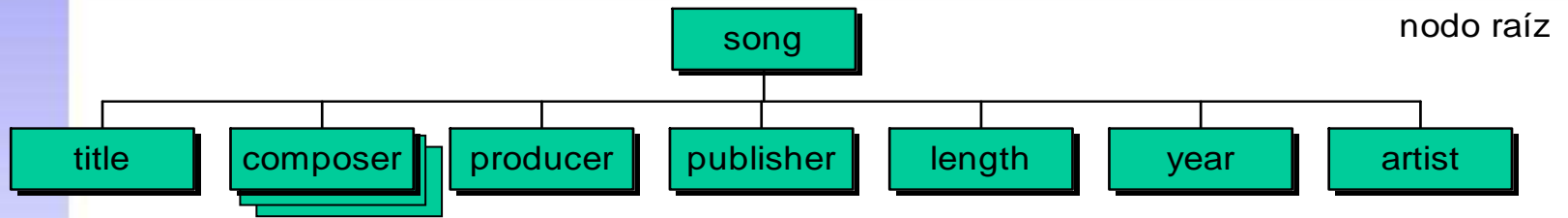






# Cómo es XML

## Representación en forma de árbol



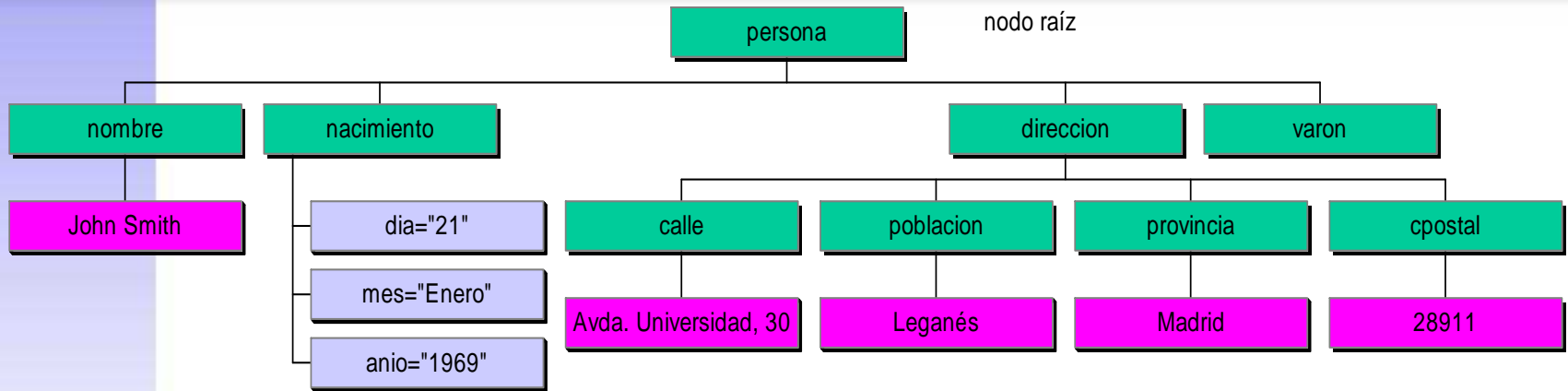
```
<?xml version='1.0'?>
<song>
  <title>Hot Cop</title>
  <composer> Jacques Morali</composer>
  <composer> Henry Beolo </composer>
  <composer> Victor Wills </composer>
  <producer> Jacques Morali </producer>
  <publisher> PolyGram </publisher>
  <length> 6:20 </length>
  <year> 1978 </year>
  <artist> Village People </artist>
</song>
```





# Cómo es XML

## Representación en forma de árbol



Nodos de elementos

Nodos hoja (contenido)

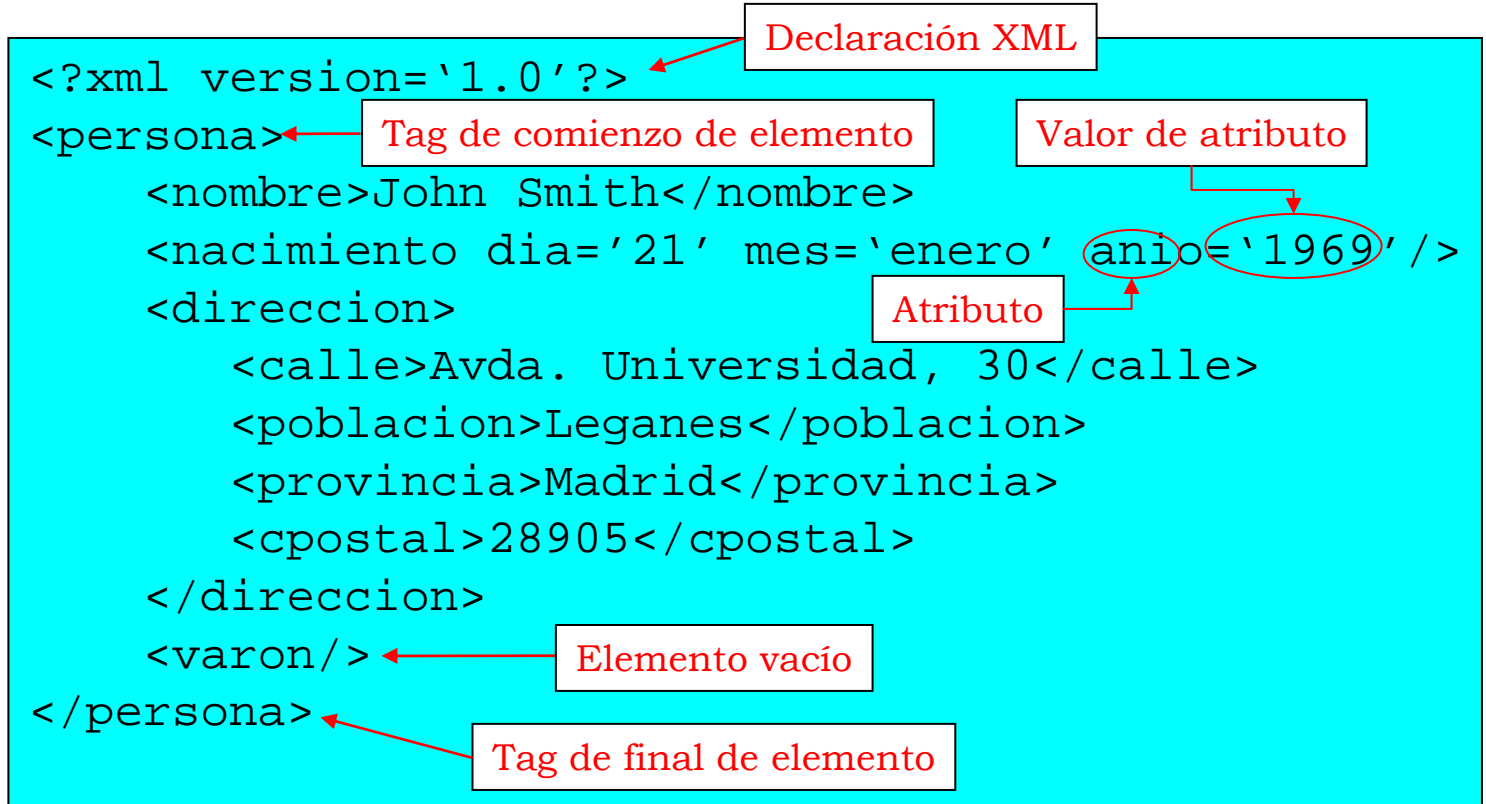
Nodos de atributos

```
<!ELEMENT persona (nombre, nacimiento?, direccion+, (varon|hembra))>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT nacimiento EMPTY>
<!ATTLIST nacimiento
    dia    CDATA    #REQUIRED
    mes    CDATA    #REQUIRED
    anio   CDATA    #REQUIRED>
<!ELEMENT direccion (calle,poblacion,provincia, cpostal)>
<!ELEMENT calle (#PCDATA)>
<!ELEMENT poblacion (#PCDATA)>
<!ELEMENT provincia (#PCDATA)>
<!ELEMENT cpostal (#PCDATA)>
<!ELEMENT varon EMPTY>
<!ELEMENT hembra EMPTY>
```





# Cómo es XML





# Cómo es XML:

## Tipos de documentos

### *Documentos bien formados (Well-formed XML)*

- Cumplen reglas de sintaxis de XML, (estructuras anidadas correctas, etc.)
- no tiene porqué cumplir una estructura predefinida

```
<?xml version="1.0" standalone="yes"?>
  <foo>
    <bar>...<blort/>...</bar>
  </foo>
```

### *Documentos válidos (Valid xml)*

- Tienen que ser bien formados
- Cumple una estructura predefinida (**DTD**)
- El parser es el encargado de comprobar la validez del documento

```
<?xml version="1.0"?>
<!DOCTYPE advert SYSTEM "http://www.foo.org/adv.dtd">
  <foo>
    <bar>...<blort/>...</bar>
  </foo>
```





# Cómo es XML:

## Compartir información

### Compartir información:

- Definir una estructura
  - **DTD's** (Document Type Definition)
  - XML **Schemas**
- Hacerla pública
  - XML **Namespaces**





# Cómo es XML (Estructura): DTD (Document Type Definition)

```
<CONTACT>
  <NAME>Roger Kaplan</NAME>
  <PHONE>555-5555</PHONE>
</CONTACT>
```

Secuencia

Alternancia

0 o 1 elementos

0 o más elementos

\*

1 o más elementos

```
<!ELEMENT CONTACT (NAME, (PHONE | EMAIL), ADDRESS?)>
  <!ELEMENT NAME (#PCDATA)>
  <!ELEMENT EMAIL (#PCDATA)>
  <!ELEMENT PHONE (#PCDATA)>
  <!ELEMENT ADDRESS (STREET+, CITY, STATE, ZIP, COUNTRY?)>
    <!ELEMENT STREET (#PCDATA)>
    <!ELEMENT CITY (#PCDATA)>
    <!ELEMENT STATE (#PCDATA)>
    <!ELEMENT ZIP (#PCDATA)>
    <!ELEMENT COUNTRY (#PCDATA)>
```





# Cómo es XML (Hojas de Estilo)

XSL: XML Stylesheet Language

***XSL = XSLT + XSL-FO's (+ XPath)***

La especificación de XSL consta de dos partes:

- **XSLT**: *transformación* inicial a otro modelo estructural (utiliza **XPath**)
- **XSL-FO**: aplicación de *formato*

XSLT utiliza otro lenguaje:

- **XPath**: Mecanismo de *localización* capaz de direccionar los elementos dentro de la estructura del documento





# Cómo es XML (Hojas de Estilo)

## XSL: XML Stylesheet Language

**XSL**

**Lenguaje de Transformación (XSLT)**

Pattern (Localización de patrones con **XPATH**)

Template (Aplicación de una plantilla al patrón seleccionado)

**Vocabulario de Formato (XSL-FO)**

Objetos de formato (Son uds tipográficas ej character)

Propiedades { Atributo: (Ej: text-decoration)  
Valor: (Ej:underline)

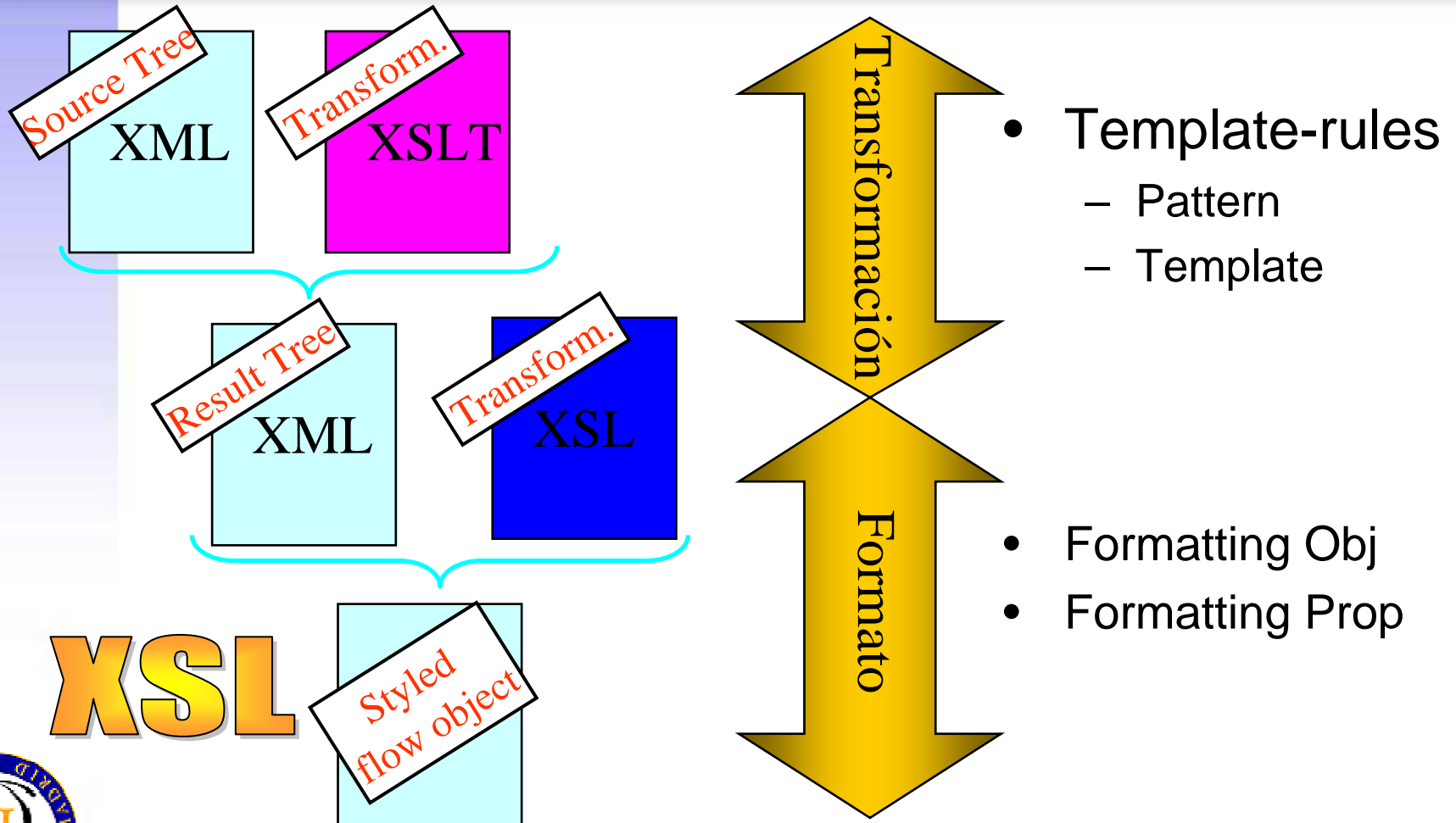






# Cómo es XML (Hojas de Estilo)

XSL: Transformación + Formato



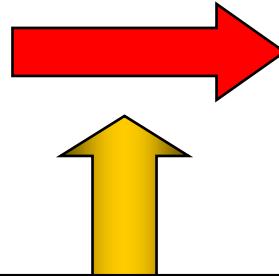


# Cómo es XML (Hojas de Estilo)

## XSL plantillas (template-rules)

*Queremos convertir el fichero document.xml en el fichero html que se muestra a continuación usando una hoja de estilo.*

```
<?xml version='1.0'?>
<document>
  Este es mi primer documento
</document>
```



```
<HTML>
  <p>
    Este es mi primer documento
  </p>
</HTML>
```

```
<xsl:template match = "document">
  <HTML>
    <p>
      <xsl:apply-templates/>
    </p>
  </HTML>
</xsl:template>
```



**xt fichero.xml fichero.xsl fichero.html**

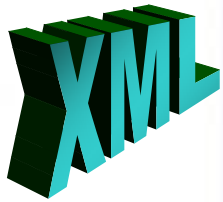


# Cómo es XML (Hojas de Estilo)

## Beneficios de XSL

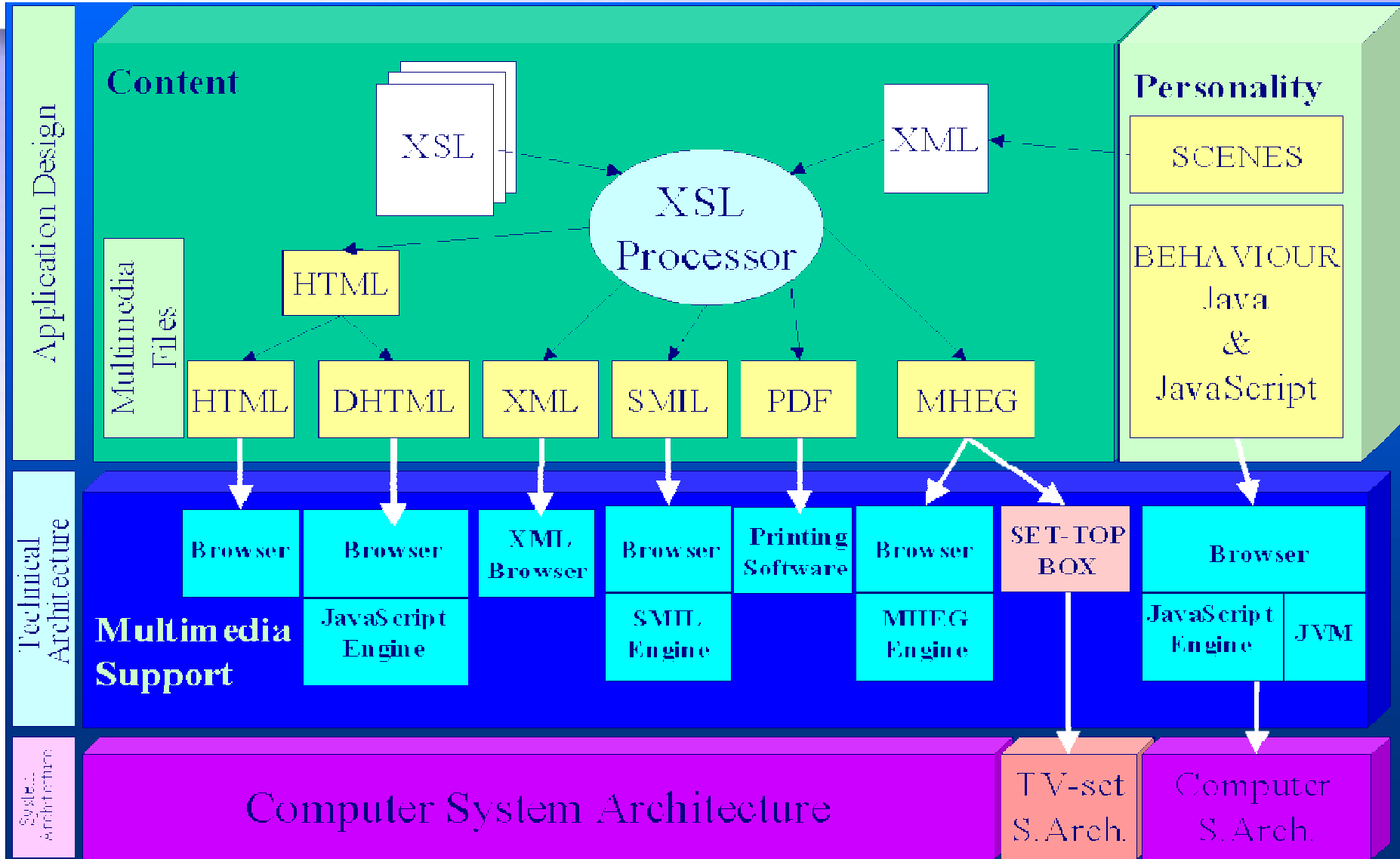
- Imprescindible para presentar información: XML no tiene información de formato
- Beneficios hojas de estilo:
  - **Centralizar** la forma de presentación
  - **Separar** la estructura del contenido
    - Reutilizar datos
    - Diferentes formatos de salida
    - Uniformizar el estilo de presentación
    - Separar la creación de contenido
    - Estilo declarativo vs procedural





# Cómo es XML (Hojas de Estilo)

## Beneficios de XSL





# Cómo es XML (Enlaces)

XLL: XML Linking Language

***XLL = XLink + XPointer (+ XPath)***

Consta de dos especificaciones

- **XLink**: define la forma en la que los documentos deben enlazarse
- **XPointer**: Describe como se debe apuntar a un lugar específico e un determinado documento XML (utiliza **XPath**)

***XLink determina el documento que se va a enlazar y XPointer marca el lugar exacto de dicho documento***





# Cómo es XML (Enlaces)

## Beneficios de XLL

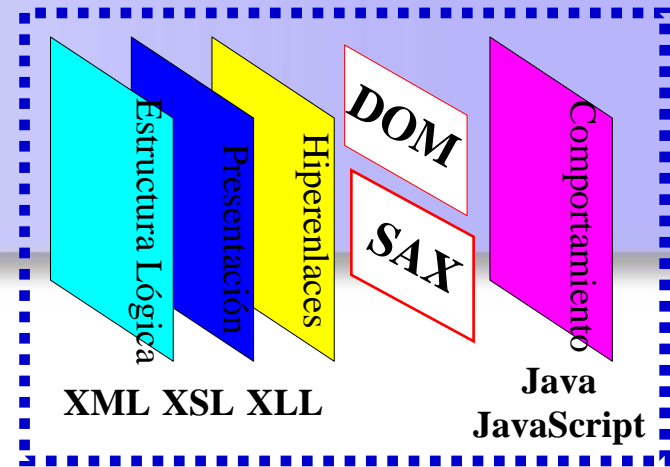
- “Alias”
- Gestión de enlaces
- Enlaces multidireccionales
- Un origen y varios destinos
- Enlaces agregados  
(Varios orígenes un sólo destino)
- Transclusión
- Atributos





# Cómo es XML

## Procesamiento



### – **Orientado a objetos (ej:DOM)**

- API que proporciona acceso al documento completo
- Se construye un árbol en memoria con la jerarquía de elementos

### – **Orientado a eventos (ej:SAX)**

- El documento se procesa de manera secuencial.
- Cada elemento en el flujo de datos dispara un evento.

***En las aplicaciones con restricciones sobre la memoria se suele utilizar el modelo orientado a eventos (ej. SAX)***



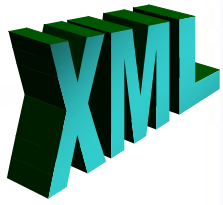


# Conclusiones

- Estructura de datos y mucho más
- No reemplazará a HTML
- Ya existen bastantes herramientas
- Cada comunidad un estándar.







# Referencias

- <http://www.w3c.org/>
- <http://xml.coverpages.org/>
- <http://www.xml.org>
- <http://www.xml.com>
- [http://www.oasis\\_open.org/cover](http://www.oasis_open.org/cover)
- <http://www.gca.org>
- <http://www.xmlsoftware.com>
- <http://www.wdvl.com/xml>
- <http://www.xmlshockwave.com>
- <http://www.alphaworks.ibm.com>
- <http://www.microsoft.com/xml>
- <http://www.it.uc3m.es/~xml>

