

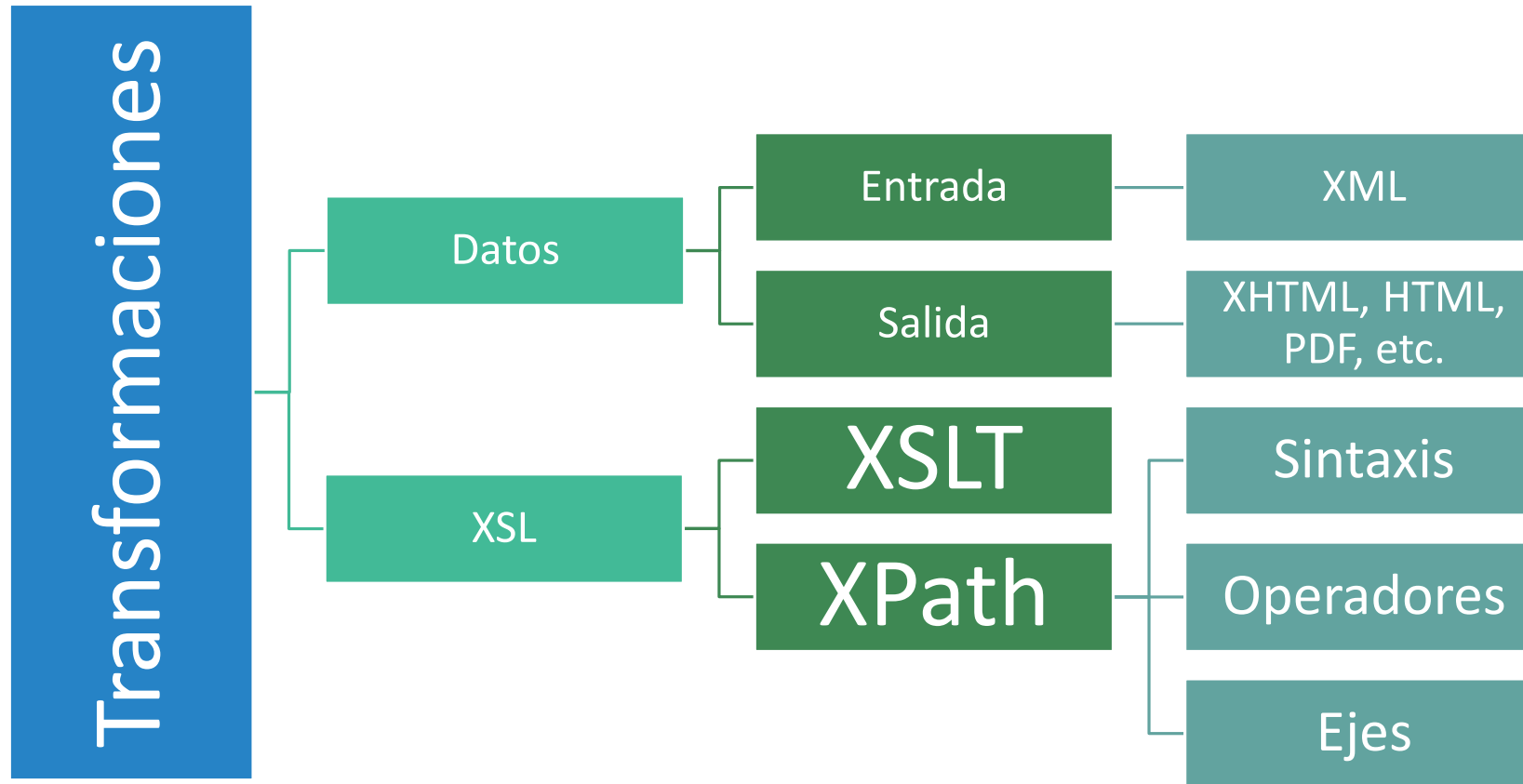


Transformaciones XSL

JUAN CARLOS CONDE RAMÍREZ

WEB-TECHNOLOGIES

Vista general



XHTML I

- XHTML v1.0 es una **reformulación de HTML4** escrito de forma que cumpla las **normas sintácticas de XML**.
- Dado que XHTML es una **aplicación de XML**, ha sido diseñado para ser ampliable.
- XHTML ha sido diseñado pensando en la **portabilidad** y su **visualización** en diferente hardware (computadoras personales, PDAs, teléfonos celulares, etc.).

XHTML II

- Los nombres de las etiquetas de elementos tienen que estar en minúscula.
- Los valores de los atributos deben estar rodeados de comillas (") o (')
- Todos los elementos tienen que estar cerrados, ya tengan contenido (`<p>...</p>`) o no (`
`).

XHTML III

- Los elementos deberán estar correctamente anidados.
- Los valores de atributos iguales sin variantes no pueden ser simplificados.

<code><OPTION VALUE="valor" SELECTED></code>	(incorrecto)
<code><option value="valor" selected="selected"></code>	(correcto)

- Algunos elementos (`html`, `body`, `head`, ...) son obligatorios.

XHTML IV

- Se debe incluir una Declaración de Tipo de Documento (DTD).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/frameset.dtd">
```



Validación
estricta

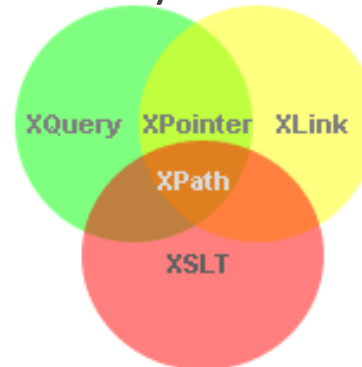
XHTML V

- El contenido de los elementos `<script>` y `<style>` tiene que ser una sección CDATA.

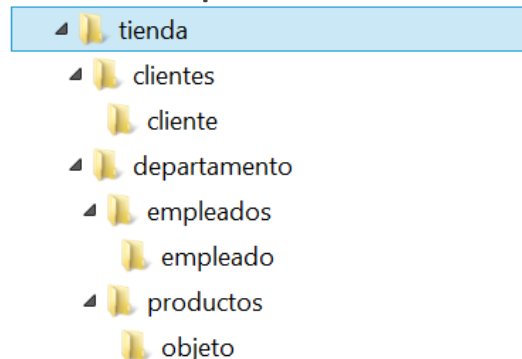
```
<script language="JavaScript">
  <!--
    <![CDATA[
      document.write("<p>Texto de prueba</p>");
    ]]>
  //-->
</script>
```

Especificación de patrones (XPath), I

- XPath es intersección de varias tecnologías XML que involucran la búsqueda de datos. Es utilizado para navegar a través de **elementos** y **atributos** en un documento XML.



- XPath usa **expresiones en forma de “ruta”** para seleccionar nodos o conjuntos de nodos en un documento XML.



Especificación de patrones (XPath), I

- Algunos de los elementos sintácticos más utilizados son:

- *nodename* Selecciona todos los nodos con ese nombre
- / Selecciona el nodo "hijo" inmediato partiendo de la raíz.
- // Selecciona los nodos descendientes del nodo actual, a "cualquier profundidad".
- . Selecciona el nodo actual.
- .. Selecciona el nodo padre del nodo actual.
- @ Selecciona un atributo.
- * Hace coincidir cualquier nodo
- @* Hace coincidir cualquier atributo.

Especificación de patrones (XPath), II

- Selecciona todos los nodos `cliente` de `/tienda/clientes`.

`/tienda/clientes/cliente`

- Selecciona todos los elementos `empleado` que aparezcan a cualquier nivel de profundidad.

`//empleado`

- Selecciona todos los elementos `empleado` que aparezcan a un nivel de profundidad de distancia por debajo de elementos `tienda`.

`/tienda/departamento/*/empleado`

Especificación de patrones (XPath), III

- Selecciona todos los atributos `precio` de los elementos `objeto` por debajo del contexto actual (o primera rama).

```
./objeto/@precio
```

- Selecciona todos los elementos `empleado` que aparezcan a un nivel de profundidad por debajo de elementos `departamento` y cuyo atributo `nombre` sea igual a *Alfredo*.

```
/tienda/departamento/*/empleado[@nombre="Alfredo"]
```

- Selecciona todos los elementos `objeto`, cuyo atributo `precio` sea menor que 500, y que el contenido de su sub-elemento `tipo` sea 1.

```
//objeto[(@precio < 500) and (tipo = 1)]
```

Especificación de patrones (XPath), IV

- Para una referencia completa de la **sintaxis** visitar:
 - https://www.w3schools.com/xml/xpath_syntax.asp
- Para una referencia más completa de los **operadores** visitar:
 - https://www.w3schools.com/xml/xpath_operators.asp
- Para una referencia completa de los **ejes de relación** visitar:
 - https://www.w3schools.com/xml/xpath_axes.asp

Hojas de estilo – XSL, I

- El *eXtended Stylesheet Language* (XSL) es un lenguaje que nos permite definir una presentación o formato diferente para un documento XML.
- Un mismo documento XML puede tener varias hojas de estilo XSL que lo muestren en diferentes formatos (HTML, PDF, RTF, PostScript, sonido, etc.).
- La aplicación de una hoja de estilo XSL a un documento XML puede ocurrir desde el origen (por ejemplo, un script en el servidor que convierta de XML a HTML)

Hojas de estilo – XSL, II

- Básicamente, XSL es un lenguaje que define una transformación entre un documento XML de entrada, y otro documento de salida.
- Una hoja de estilo XSL es una **serie de reglas** que determinan cómo va a ocurrir la transformación.
- Cada regla se compone de un patrón (*pattern*) y una acción o plantilla (*template*).

XSL = XSLT + Xpath

- Esta familia está formada por tres lenguajes:
 - **XSLT** (*Extensible Stylesheet Language Transformations*) es lenguaje de transformación para hojas de estilo; permite convertir documentos XML de una sintaxis a otra.
 - **XSL-FO** (*Extensible Stylesheet Language - Formatting Objects*), permite especificar el formato visual con el cual se quiere presentar un documento XML, es usado principalmente para generar documentos PDF.
 - **Xpath** (*XML Path Language*) es una sintaxis (no basada en XML) para acceder o referirse a porciones de un documento XML.

XSL = XSLT + Xpath

- **XSLT** (XSL Transformations)

- Define las acciones o transformaciones a realizar.
- W3C Recommendation 16 noviembre 1999
- <http://www.w3.org/TR/xslt>

- **Xpath**

- Localización de elementos en el documento.
- W3C Recommendation 16 noviembre 1999
- <http://www.w3.org/TR/xpath>

Ejemplo: XML (origen), I

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="04_links.xsl" type="text/xsl"?>
<links>
  <header title="Mis Sitios Favoritos"/>
  <item href="https://www.youtube.com" title="YouTube"/>
  <item href="http://www.milenio.com" title="Diario Milenio"/>
  <item href="https://www.w3schools.com" title="W3Schools"/>
  <item href="http://climate.cs.buap.mx/CondeJC/cursos" title="Climate"/>
</links>
```

Ejemplo: XSL (destino), II

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html" encoding="UTF-8" doctype-public="-//W3C//DTD XHTML 1.0 Strict//EN" doctype-
system="http://www.w3.org/TR/xhtml1/DTD/strict.dtd"/>
  <xsl:template match="/">
    <html>
      <head>
        <title>Mi primera transformación</title>
      </head>
      <body>
        <h2><xsl:value-of select="//header/@title"/></h2>
        <ul>
          <xsl:for-each select="links/item">
            <li>
              <a>
                <xsl:attribute name="href">
                  <xsl:value-of select="./@href"/>
                </xsl:attribute>
                <xsl:value-of select="./@title"/>
              </a>
            </li>
          </xsl:for-each>
        </ul>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Definición de reglas, I

```
<xsl:stylesheet></xsl:stylesheet>
```

- Es el elemento raíz de la hoja de estilo. Contiene elementos `xml:template` y `xml:script`.

```
<xsl:template match="pattern"></xsl:template>
```

- Define un patrón para trabajar sobre determinado conjunto de nodos, de un cierto tipo o en un contexto determinado. Es una expresión XPath, si se usa "/" se hace referencia a todo el documento de origen.

```
<xsl:apply-template select="pattern"/>
```

- Indica al procesador que busque el *template* adecuado para el tipo y contexto de cada nodo seleccionado.

Definición de reglas, II

`<xsl:value-of select="pattern"/>`

- Inserta el valor del nodo seleccionado como texto.

`<xsl:attribute name="nombre"> valor </xsl:attribute>`

- Crea un atributo y lo añade al elemento de salida.

`<xsl:element name="nombre"></xsl:element>`

- Crea un elemento con el nombre indicado. Esto es útil para evitar conflictos de nombres.

Elementos XSLT III

```
<xsl:for-each order-by="criterio" select="pattern">
```

- Repite un template de forma iterativa.

```
<xsl:if test="expresion booleana"></xsl:if>
```

- Realiza un test, y ejecuta un *template* condicionalmente.

```
<xsl:choose>  
  <xsl:when test=" expresion1 "> ... </xsl:when>  
  <xsl:when test=" expresion2 "> ... </xsl:when>  
  ...  
  <xsl:otherwise> ... </xsl:otherwise>  
</xsl:choose>
```

- Esta construcción permite verificar condiciones múltiples.

Métodos en XSLT

- XSLT provee una serie de métodos que pueden ser llamados desde los elementos:

`<xsl:script/>` y `<xsl:eval/>`

- Para una referencia más completa visitar el siguiente sitio:

https://www.w3schools.com/xml/xsl_intro.asp