

XSLT (I)

TRANSFORMACIÓN DE DOCUMENTOS XML

2º DAM

HOJAS DE ESTILO XSLT

XSLT es un lenguaje declarativo.

XSLT es a XML lo que las hojas de estilo en cascada CSS a HTML, permite tomar pleno control sobre los datos, pudiendo establecerse criterios como qué datos ver, en qué orden visualizarlos, estableciendo filtros y definiendo formatos de salida para su representación.

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

HOJAS DE ESTILO XSLT

La transformación del documento se realiza de la siguiente manera:

- El procesador analiza el documento y construye el árbol del documento.
- El procesador recorre el árbol del documento desde el nodo raíz.
- En cada nodo recorrido, el procesador aplica o no alguna plantilla:
 - Si a un nodo no se le puede aplicar ninguna plantilla, su contenido se incluye en el documento final (el texto del nodo, no el de los nodos descendientes). A continuación, el procesador recorre sus nodos hijos.
 - Si a un nodo se le puede aplicar una plantilla, se aplica la plantilla. La plantilla puede generar texto que se incluye en el documento final. En principio, el procesador no recorre sus nodos hijos, salvo que la plantilla indique al procesador que sí que deben recorrerse los nodos hijos.
- Cuando el procesador ha recorrido el árbol, se ha terminado la transformación.

DECLARACIÓN HOJA DE ESTILOS

Una hoja de estilo XSLT es un documento XML con extensión .xslt que contiene al menos las etiquetas:

- la declaración xml `<?xml>`, propia de cualquier documento XML.
- la instrucción `<xsl:stylesheet>` es la etiqueta raíz de la hoja de estilo, sus atributos indican la versión y el espacio de nombres correspondiente.

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">  
</xsl:stylesheet>
```

DECLARACIÓN HOJA DE ESTILOS

Dentro de la instrucción **<xsl:stylesheet>** se pueden encontrar los llamados elementos de alto nivel y las plantillas.

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

  <xsl:output method="xml" indent="yes"/>

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

</xsl:stylesheet>
```

← Tipo de salida producida.

← La plantilla:

- **match** → Indica los elementos afectados por la plantilla y contiene una expresión XPath.
- El contenido de la instrucción define la transformación a aplicar (si la instrucción no contiene nada sustituirá el nodo por nada, es decir, eliminará el nodo, aunque conservará el texto contenido en el elemento).

DECLARACIÓN HOJA DE ESTILOS

Cuando se aplica una plantilla a un nodo, en principio no se recorren los nodos descendientes.

Para indicar que sí queremos recorrer los nodos descendientes y aplicarles las plantillas que les correspondan, hay que utilizar la instrucción

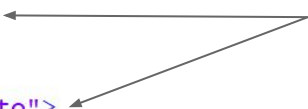
<xsl:apply-templates />

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

  <xsl:template match="/">
    <xsl:apply-templates />
  </xsl:template>

  <xsl:template match="elemento">
  </xsl:template>


</xsl:stylesheet>
```



ENLAZAR DOCUMENTOS XML CON HOJAS DE ESTILO XSLT

Se puede asociar de forma permanente una hoja de estilo XSLT a un documento XML mediante la instrucción de procesamiento **<?xml-stylesheet ?>**, la misma que permite asociar hojas de estilo CSS.

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="plantilla-xslt.xml"?>
```



Ruta y nombre de la
plantilla de xslt

NAVEGADORES WEB - ¡CUIDADO!

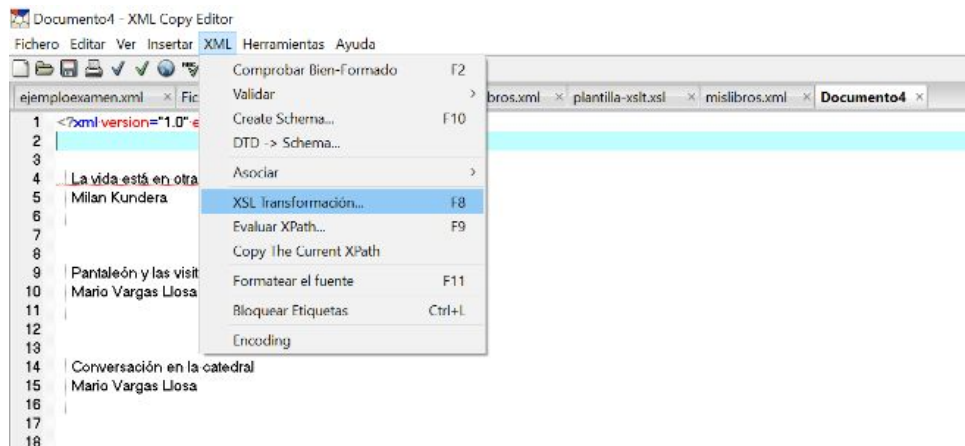
Podemos abrir nuestro .xml para comprobar la salida pero...

Google Chrome y las nuevas versiones de otros navegadores sólo muestran el documento transformado si se ha servido desde un servidor web (http://...). Si el documento se abre directamente como un archivo local (file://...), Google Chrome muestra un página en blanco.

Con Internet Explorer sí podrías ver la salida al aplicar la hoja de estilos.

XML COPY EDITOR

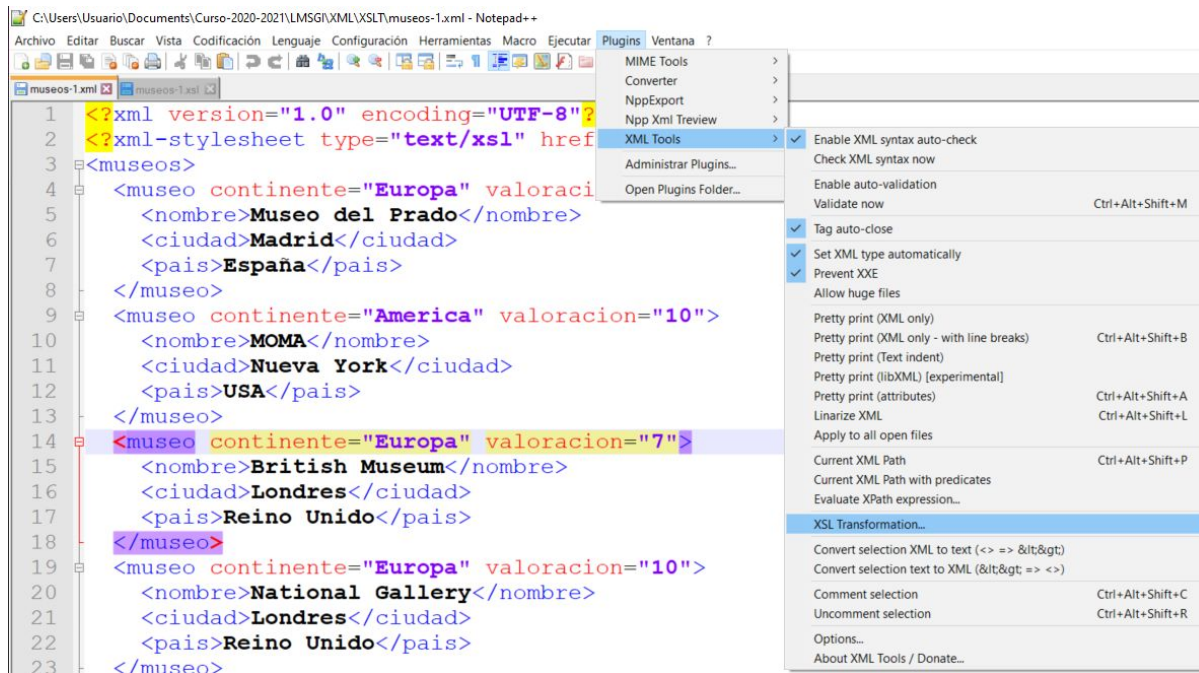
Cuando queramos visualizar el xml resultante: Menú → XML → XSL Transformación.



Si utilizais Notepad++ la visualización puede verse distinta.

NOTEPAD ++

Si tenemos el plugin de XML también podemos transformar la salida.



EJEMPLO A SEGUIR

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="plantilla-xslt.xsl"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

← Plantilla XSLT

PLANTILLAS VACÍAS O NO EXISTENTES

Si no hay plantillas, el procesador simplemente recorre todos los nodos y extrae el texto contenido por cada nodo.

Como la plantilla está vacía visualiza el título y autor.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

</xsl:stylesheet>
```

Resultado

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

La vida está en otra parte
Milan Kundera

Pantaleón y las visitadoras
Mario Vargas Llosa

Conversación en la catedral
Mario Vargas Llosa
```

PLANTILLAS VACÍAS O NO EXISTENTES

Si hay una plantilla vacía asociada a un elemento imprime ese nodo pero no recorre los nodos hijos.

Ejemplo: Match en autor, imprime autor.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

  <xsl:template match="autor">
  </xsl:template>

</xsl:stylesheet>
```

Resultado

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

  La vida está en otra parte

  Pantaleón y las visitadoras

  Conversación en la catedral
```

PLANTILLAS VACÍAS O NO EXISTENTES

Si la plantilla vacía se aplica al nodo raíz, el procesador no genera ningún resultado en el documento final ni recorre ningún nodo hijo. La salida es vacía.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

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

</xsl:stylesheet>
```

Resultado

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

INSTRUCCIÓN <XSL:VALUE-OF>

<xsl:value-of> extrae el contenido del nodo seleccionado.

Ejemplo: Se aplica una plantilla a libro y se selecciona el valor de autor.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

  <xsl:template match="libro">
    <xsl:value-of select="autor"/>
  </xsl:template>

</xsl:stylesheet>
```

Resultado

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

Milan Kundera
Mario Vargas Llosa
Mario Vargas Llosa
```

INSTRUCCIÓN <XSL:VALUE-OF>

Ejemplo: Se obtienen el título y el autor, pero uno a continuación de otro. Los saltos de línea se crean tras cada aplicación de la regla (es decir, a cada libro), pero no en el interior de la regla.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

  <xsl:template match="libro">
    <xsl:value-of select="titulo"/>
    <xsl:value-of select="autor"/>
  </xsl:template>

</xsl:stylesheet>
```

Resultado

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

La vida está en otra parteMilan Kundera
Pantaleón y las visitadorasMario Vargas Llosa
Conversación en la catedralMario Vargas Llosa
```


INSTRUCCIÓN <XSL:VALUE-OF>

Ejemplo: Los autores se obtienen gracias a la regla que extrae el contenido del nodo (el carácter punto "." hace referencia al propio elemento) y los títulos se obtienen porque al no haber reglas para ese nodo se extrae el contenido.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

  <xsl:template match="autor">
    <xsl:value-of select="."/>
  </xsl:template>

</xsl:stylesheet>
```

Resultado

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

La vida está en otra parte
Milan Kundera

Pantaleón y las visitadoras
Mario Vargas Llosa

Conversación en la catedral
Mario Vargas Llosa
```

INSTRUCCIÓN <XSL:VALUE-OF>

Ejemplo: También se pueden extraer los valores de los atributos, utilizando @. Las fechas de publicación se obtienen gracias a la regla que extraen el valor del atributo y los títulos y autores se obtienen porque al no haber reglas para ese nodo se extrae el contenido.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">

  <xsl:template match="fechaPublicacion">
    <xsl:value-of select="@año"/>
  </xsl:template>

</xsl:stylesheet>
```

Resultado

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

La vida está en otra parte
Milan Kundera
1973

Pantaleón y las visitadoras
Mario Vargas Llosa
1973

Conversación en la catedral
Mario Vargas Llosa
1969
```

GENERAR TEXTO ADICIONAL

Podemos añadir texto escribiéndolo en la regla. Por ejemplo, HTML. Aunque sin las etiquetas `<html></html>` el navegador no lo formatea como web.

Ejemplo: Autores con párrafo.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

  <xsl:template match="libro">
    <p><xsl:value-of select="autor"/></p>
  </xsl:template>

</xsl:stylesheet>
```

Resultado

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

<p>Milan Kundera</p>
<p>Mario Vargas Llosa</p>
<p>Mario Vargas Llosa</p>
```

GENERAR TEXTO ADICIONAL

Podemos añadir texto escribiéndolo en la regla. Por ejemplo, HTML.

Ejemplo: Autores y títulos con párrafo.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

  <xsl:template match="libro">
    <p><xsl:value-of select="autor"/></p>
    <p><xsl:value-of select="titulo"/></p>
  </xsl:template>
</xsl:stylesheet>
```

Resultado

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

<p>Milan Kundera</p>
<p>La vida está en otra parte</p>

<p>Mario Vargas Llosa</p>
<p>Pantaleón y las visitadoras</p>

<p>Mario Vargas Llosa</p>
<p>Conversación en la catedral</p>
```

GENERAR TEXTO ADICIONAL

El procesador XSLT sólo aplica una regla a cada nodo. Si tenemos dos reglas para el mismo nodo, el procesador sólo aplica una de ellas (la última, en este caso).

Ejemplo : Solo visualiza los libros.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

  <xsl:template match="libro">
    <p><xsl:value-of select="autor"/></p>
  </xsl:template>

  <xsl:template match="libro">
    <p><xsl:value-of select="titulo"/></p>
  </xsl:template>

</xsl:stylesheet>
```

Resultado

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

<p>La vida está en otra parte</p>

<p>Pantaleón y las visitadoras</p>

<p>Conversación en la catedral</p>
```

GENERAR TEXTO ADICIONAL

El procesador XSLT sólo aplica una regla a cada nodo. Si tenemos dos reglas para el mismo nodo, el procesador sólo aplica una de ellas (la última, en este caso).

Ejemplo: Solo visualiza los autores.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

  <xsl:template match="libro">
    <p><xsl:value-of select="titulo"/></p>
  </xsl:template>

  <xsl:template match="libro">
    <p><xsl:value-of select="autor"/></p>
  </xsl:template>

</xsl:stylesheet>
```

Resultado

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

<p>Milan Kundera</p>

<p>Mario Vargas Llosa</p>

<p>Mario Vargas Llosa</p>
```

GENERAR TEXTO ADICIONAL

Además de generar etiquetas, se puede generar texto.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

  <xsl:template match="libro">
    <p><xsl:value-of select="autor"/> escribió
    "<xsl:value-of select="titulo"/>"</p>
  </xsl:template>

</xsl:stylesheet>
```

Resultado

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

<p>Milan Kundera escribió "La vida está en otra
parte"</p>

<p>Mario Vargas Llosa escribió "Pantaleón y las
visitadoras"</p>

<p>Mario Vargas Llosa escribió "Conversación en
la catedral"</p>
```

APLICAR REGLAS A SUBNODOS: <XSL:APPLY-TEMPLATES>

<xsl:apply-templates> hace que se apliquen a los subelementos las reglas que les sean aplicables.

Ejemplo: Se genera la etiqueta <html> además de unos párrafos con los nombres de los autores.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

  <xsl:template match="/">
    <html>
      <h1>Autores</h1>
      <xsl:apply-templates />
    </html>
  </xsl:template>

  <xsl:template match="libro">
    <p><xsl:value-of select="autor" /></p>
  </xsl:template>

</xsl:stylesheet>
```

Resultado

```
<?xml version="1.0" encoding="UTF-8"?>
<html><h1>Autores</h1>
  <p>Milan Kundera</p>
  <p>Mario Vargas Llosa</p>
  <p>Mario Vargas Llosa</p>
</html>
```

La primera regla sustituye el elemento raíz (y todos sus subelementos) por las etiquetas <html> y <h1>, pero además aplica a los subelementos las reglas que les son aplicables. En este caso, sólo hay una regla para los elementos <libro> que generan los párrafos.

SALTOS DE LÍNEA Y ESPACIOS EN BLANCO

Al transformar un documento, los procesadores XSLT incorporan saltos de línea y espacios en blanco en el resultado, pero no lo hacen de forma uniforme.

Por ejemplo, XML Copy Editor y Notepad++ (con el plug-in XML Tools) producen diferentes resultados.

No parece haber una solución sencilla que funcione en todos los procesadores, pero sí soluciones que funcionen en cada uno de ellos.

<XSL:STRIP-SPACE>

<xsl:strip-space> permite indicar si los elementos que contienen únicamente espacios en blanco se incluyen en la transformación o no.

En el caso de XML Copy Editor, la forma más sencilla de mejorar el formato de presentación de los resultados, eliminando líneas en blanco innecesarias y sangrando los elementos anidados, es utilizar la instrucción `<xsl:strip-space>`. Pero debe tenerse en cuenta que esta instrucción no produce el mismo resultado en otros procesadores XSLT (como en Notepad++ con XML Tools).

<XSL:STRIP-SPACE>

Ejemplo: Antes la etiqueta <h1> se generaba en la misma línea que la etiqueta <html>, pero en el ejemplo siguiente se generan en líneas distintas (y las etiquetas se muestran sangradas) al utilizar la instrucción <xsl:strip-space>.

XSLT


```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:strip-space elements="*" />

  <xsl:template match="/">
    <html>
      <h1>Autores</h1>
      <xsl:apply-templates />
    </html>
  </xsl:template>

  <xsl:template match="libro">
    <p><xsl:value-of select="autor"/></p>
  </xsl:template>
</xsl:stylesheet>
```

Resultado

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <h1>Autores</h1>
  <p>Milan Kundera</p>
  <p>Mario Vargas Llosa</p>
  <p>Mario Vargas Llosa</p>
</html>
```



<XSL:ATTRIBUTE>

<xsl:attribute> permite generar un atributo y su valor. Se utiliza cuando el valor del atributo se obtiene a su vez de algún nodo.

Tenemos el siguiente XML y queremos que el contenido de imagen se guarde en un atributo:

```
<?xml version="1.0" encoding="UTF-8"?>
<licencias>
  <licencia>
    <nombre>Creative Commons By - Share Alike</nombre>
    <imagen>cc-bysa-88x31.png</imagen>
  </licencia>
</licencias>
```

<XSL:ATTRIBUTE>

Ejemplo: En la hoja de estilo XSLT, la etiqueta se escribe con apertura y cierre. Faltaría la etiqueta <html> para que lo interprete.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">

  <xsl:template match="licencia">
    <p><img>
      <xsl:attribute name="src">
        <xsl:value-of select="imagen" />
      </xsl:attribute>
    </img>
    </p>
  </xsl:template>

</xsl:stylesheet>
```

Resultado

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

<XSL:ATTRIBUTE>

Ejemplo: Aquí el mismo ejemplo con la etiqueta <html>.

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">

  <xsl:template match="/">
    <html>
      <xsl:apply-templates />
    </html>
  </xsl:template>

  <xsl:template match="licencia">
    <p><img>
      <xsl:attribute name="src">
        <xsl:value-of select="imagen" />
      </xsl:attribute>
    </img>
    </p>
  </xsl:template>

</xsl:stylesheet>
```



Resultado

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