

## XSLT: Transformaciones XSL

Fuente: <https://www.mclibre.org/consultar/xml/lecciones/xml-xslt.html>

Con XML Copy Editor elaboramos el documento “biblioteca.xml”:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="ejemplo_01_plantillaVacia.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>
```

Observamos que tiene asociada de forma permanente la hoja de estilo XSLT “ejemplo\_01\_plantillaVacia.xsl”.

NOTA: En XML Copy Editor se puede asociar mediante menú XML – Asociar (...); en caso de no tener hoja de estilo asociada, cuando invocamos ejecución de transformación (F8) nos solicita la selección del archivo XSL para realizar dicha transformación.

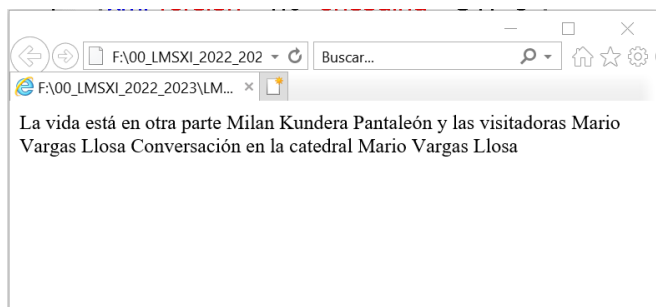
Para el ejemplo anterior, observamos que el archivo “ejemplo\_01\_plantillaVacia.xsl” consiste en una transformación que está vacía, no dispone de plantillas a ejecutar:

Archivo XSL
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;  &lt;/xsl:stylesheet&gt;</pre>

Resultado en XML Copy Editor
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;  La vida está en otra parte Milan Kundera  Pantaleón y las visitadoras Mario Vargas Llosa  Conversación en la catedral Mario Vargas Llosa</pre>

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

Ejecutamos el archivo “biblioteca.xml” en un navegador:



observamos el resultado (no respeta espacios ni saltos de línea)

## Más ejemplos...

Para realizar los ejemplos siguientes desvinculamos, de forma permanente, la hoja de estilo en el documento biblioteca.xml (marcamos como comentario). En las siguientes ejecutamos la transformación mediante la selección del archivo .xsl en el menú F8.

### Ejemplo-2

Archivo XSL	Resultado en XML Copy Editor
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;    &lt;xsl:template match="autor"&gt;   &lt;/xsl:template&gt;  &lt;/xsl:stylesheet&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;  La vida está en otra parte  Pantaleón y las visitadoras  Conversación en la catedral</pre>

El resultado incluye el contenido de los nodos <titulo> ya que no hay regla para ellos, los de autor no aparecen debido a que existe plantilla, aunque se trata una plantilla vacía.

### Ejemplo-3

Archivo XSL	Resultado en XML Copy Editor
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;    &lt;xsl:template match="/"&gt;   &lt;/xsl:template&gt;  &lt;/xsl:stylesheet&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;</pre>

Como la plantilla, vacía, se aplica al nodo raíz, el procesador no genera ningún resultado en el documento final ni recorre ningún nodo hijo de este.

## La instrucción <xsl:value-of>

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

### Ejemplo-4

Archivo XSL	Resultado en XML Copy Editor
-------------	------------------------------

```
<?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>
```

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

Milan Kundera  
Mario Vargas Llosa  
Mario Vargas Llosa

el documento final contiene los autores de los libros porque la plantilla los genera con la instrucción `<xsl:value-of>`. Como se ha aplicado una plantilla al nodo `<libro>`, sus hijos (`<titulo>`, `<autor>` y `<fechaPublicacion>`) no se recorren. Por eso, los títulos de los libros no aparecen en el documento final.

## Ejemplo-5

### Archivo XSL

```
<?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 en XML Copy Editor

```
<?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
```

se obtienen el título y el autor de los libros, 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.

## Ejemplo-6

### Archivo XSL

```
<?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 en XML Copy Editor

```
<?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
```

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. El resultado es el mismo que el del ejemplo 1, pero por motivos distintos.

## Ejemplo-7

También se pueden extraer los valores de los atributos, utilizando `@`.

### Archivo XSL

### Resultado en XML Copy Editor

```
<?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>
```

```
<?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

las fechas de publicación se obtienen gracias a la regla que extrae el valor del atributo y los títulos y autores se obtienen porque al no haber reglas para ese nodo se extrae el contenido.

## Generar texto adicional

### Ejemplo-8

#### Archivo XSL

```
<?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 en XML Copy Editor

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

<p>Milan Kundera</p>

<p>Mario Vargas Llosa</p>

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

se obtienen los nombres de los autores porque la regla selecciona el nodo <libro> como en el ejemplo, 2-1, pero además generamos las etiquetas <p>. El resultado sigue sin verse bien en el navegador, porque aunque hay etiquetas <p>, falta la etiqueta global <html>.

### Ejemplo-9

Dentro de la regla podemos hacer referencia a varios subnodos.

#### Archivo XSL

```
<?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 en XML Copy Editor

```
<?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>
```

se obtienen los nombres de los autores y los títulos de los libros porque la regla selecciona el nodo <libro> como en el ejemplo, 2-1, pero además generamos las etiquetas <p>. El resultado sigue sin verse bien en el navegador, porque aunque hay etiquetas <p>, falta la etiqueta global <html>

### Ejemplo-10

Archivo XSL	Resultado en XML Copy Editor
<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;    &lt;xsl:template match="libro"&gt;     &lt;p&gt;&lt;xsl:value-of select="autor"/&gt;&lt;/p&gt;   &lt;/xsl:template&gt;    &lt;xsl:template match="libro"&gt;     &lt;p&gt;&lt;xsl:value-of select="título"/&gt;&lt;/p&gt;   &lt;/xsl:template&gt;  &lt;/xsl:stylesheet&gt; </pre>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt;  &lt;p&gt;La vida está en otra parte&lt;/p&gt;  &lt;p&gt;Pantaleón y las visitadoras&lt;/p&gt;  &lt;p&gt;Conversación en la catedral&lt;/p&gt; </pre>

### y ejemplo-11

Archivo XSL	Resultado en XML Copy Editor
<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;    &lt;xsl:template match="libro"&gt;     &lt;p&gt;&lt;xsl:value-of select="título"/&gt;&lt;/p&gt;   &lt;/xsl:template&gt;    &lt;xsl:template match="libro"&gt;     &lt;p&gt;&lt;xsl:value-of select="autor"/&gt;&lt;/p&gt;   &lt;/xsl:template&gt;  &lt;/xsl:stylesheet&gt; </pre>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt;  &lt;p&gt;Milan Kundera&lt;/p&gt;  &lt;p&gt;Mario Vargas Llosa&lt;/p&gt;  &lt;p&gt;Mario Vargas Llosa&lt;/p&gt; </pre>

Los anteriores ejemplos (10 y 11) intentan conseguir el mismo resultado que el ejemplo anterior (ejemplo 3-2), pero utilizando dos reglas, y no lo consiguen.

¿Por qué en un caso se obtienen sólo los títulos y en el otro sólo los autores? Porque 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-12

Además de generar etiquetas, se puede generar texto.

Archivo XSL	Resultado en XML Copy Editor
<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;    &lt;xsl:template match="libro"&gt;     &lt;p&gt;&lt;xsl:value-of select="autor"/&gt; escribió "&lt;xsl:value-of select="título"/&gt;"&lt;/p&gt;   &lt;/xsl:template&gt;  &lt;/xsl:stylesheet&gt; </pre>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt;  &lt;p&gt;Milan Kundera escribió "La vida está en otra parte"&lt;/p&gt;  &lt;p&gt;Mario Vargas Llosa escribió "Pantaleón y las visitadoras"&lt;/p&gt;  &lt;p&gt;Mario Vargas Llosa escribió "Conversación en la catedral"&lt;/p&gt; </pre>

se generan frases a partir del contenido de los nodos.

## Aplicar reglas a subnodos: la instrucción <xsl:apply-templates>

### Ejemplo-13

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

Archivo XSL	Resultado en XML Copy Editor
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;    &lt;xsl:template match="/"&gt;     &lt;html&gt;       &lt;h1&gt;Autores&lt;/h1&gt;       &lt;xsl:apply-templates /&gt;     &lt;/html&gt;   &lt;/xsl:template&gt;    &lt;xsl:template match="libro"&gt;     &lt;p&gt;&lt;xsl:value-of select="autor"/&gt;&lt;/p&gt;   &lt;/xsl:template&gt;  &lt;/xsl:stylesheet&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;html&gt;&lt;h1&gt;Autores&lt;/h1&gt;   &lt;p&gt;Milan Kundera&lt;/p&gt;   &lt;p&gt;Mario Vargas Llosa&lt;/p&gt;   &lt;p&gt;Mario Vargas Llosa&lt;/p&gt; &lt;/html&gt;</pre>

se genera la etiqueta <html> además de unos párrafos con los nombres de los autores. Este ejemplo sí que se puede ver en el navegador ya que se interpreta como 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.

## Salto de línea y espacios en blanco: las instrucciones <xsl:text> y <xsl:strip-space>

### Ejemplo-14

Archivo XSL	Resultado en XML Copy Editor
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;   &lt;xsl:strip-space elements="*" /&gt;    &lt;xsl:template match="/"&gt;     &lt;html&gt;       &lt;h1&gt;Autores&lt;/h1&gt;       &lt;xsl:apply-templates /&gt;     &lt;/html&gt;   &lt;/xsl:template&gt;    &lt;xsl:template match="libro"&gt;     &lt;p&gt;&lt;xsl:value-of select="autor"/&gt;&lt;/p&gt;   &lt;/xsl:template&gt;  &lt;/xsl:stylesheet&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;html&gt;   &lt;h1&gt;Autores&lt;/h1&gt;   &lt;p&gt;Milan Kundera&lt;/p&gt;   &lt;p&gt;Mario Vargas Llosa&lt;/p&gt;   &lt;p&gt;Mario Vargas Llosa&lt;/p&gt; &lt;/html&gt;</pre>

Las etiquetas <h1> y <html> se generan en líneas distintas (y las etiquetas se muestran sangradas) al utilizar la instrucción <xsl:strip-space>.

## La instrucción <xsl:text>

La instrucción <xsl:text> permite generar texto que no se puede generar simplemente añadiéndolo (saltos de líneas y espacios en blanco, por ejemplo).

## Ejemplo-15

Archivo XSL	Resultado en XML Copy Editor
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;    &lt;xsl:template match="/"&gt;     &lt;html&gt;       &lt;xsl:text&gt;&amp;#10; &lt;/xsl:text&gt;       &lt;h1&gt;Autores&lt;/h1&gt;       &lt;xsl:apply-templates /&gt;     &lt;/html&gt;   &lt;/xsl:template&gt;    &lt;xsl:template match="libro"&gt;     &lt;p&gt;&lt;xsl:value-of select="autor"/&gt;&lt;/p&gt;   &lt;/xsl:template&gt;  &lt;/xsl:stylesheet&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;html&gt;   &lt;h1&gt;Autores&lt;/h1&gt;   &lt;p&gt;Milan Kundera&lt;/p&gt;   &lt;p&gt;Mario Vargas Llosa&lt;/p&gt;   &lt;p&gt;Mario Vargas Llosa&lt;/p&gt; &lt;/html&gt;</pre>

La etiquetas <h1> y <html> se generan en líneas distintas al añadir un salto de línea con la entidad de carácter &#10; (y un par de espacios para alinear la etiqueta <h1> con las etiquetas <p>)