



LENGUAJES DE MARCAS

UD7. Transformaciones con XSLT - Parte 1

Desarrollo de Aplicaciones Web

Profesores: Diana Bautista / Óscar Villar

Índice de contenido

- 
1. Introducción a XSLT
 2. Hojas de estilo
 3. Enlazar documentos XSLT en XML y abrirlos en el navegador
 4. Ejemplos de plantillas XSLT

1. Introducción a XSLT

Introducción a XSLT

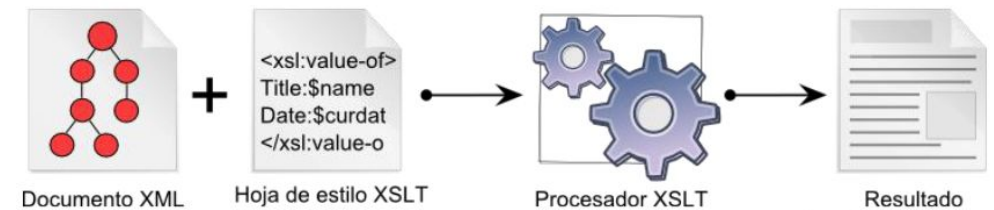
XSLT (Transformaciones XSL) es un lenguaje de programación declarativo que permite generar documentos a partir de documentos XML

La hoja de estilo **XSLT** es el documento que **contiene el código fuente del programa**, es decir, las **reglas de transformación** que se van a aplicar al documento inicial.

El **procesador XSLT** es el programa de ordenador que aplica al documento inicial las reglas de transformación incluidas en la hoja de estilo XSLT y genera el documento final.

El resultado de la ejecución del programa es un nuevo documento (que puede ser un documento XML o no)

A un documento XML se le pueden aplicar diferentes hojas de estilo XSLT para obtener diferentes resultados



2. Hojas de estilo XSLT

Hojas de estilo XSLT

Una hoja de estilo XSLT debe contener al menos las siguientes 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.

Dentro de la instrucción `<xsl:stylesheet>` nos encontramos **los elementos de alto nivel o plantillas**:

- El elemento de alto nivel `<xsl:output>` indica el tipo de salida producida.
- La instrucción `<xsl:template>` es una plantilla.

```
<?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" version="4.01" encoding="UTF-8"/>
<xsl:template match="/">
</xsl:template>
</xsl:stylesheet>
```

Hojas de estilo XSLT

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 />**, como en el ejemplo siguiente:

```
<?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>
```

3. Enlazar hojas de estilo y abrir en el navegador

Enlazar hojas de estilo y abrir

Se puede asociar de forma permanente una hoja de estilo XSLT a un documento XML al igual que hacemos con las hojas de estilo en HTML.

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

Cuando se visualiza en un navegador web un documento XML enlazado con una hoja de estilo XSLT, los navegadores muestran el resultado de la transformación, aunque si se muestra el código fuente de la página, los navegadores muestran el documento XML original.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="ejercicio2.xsl"?>

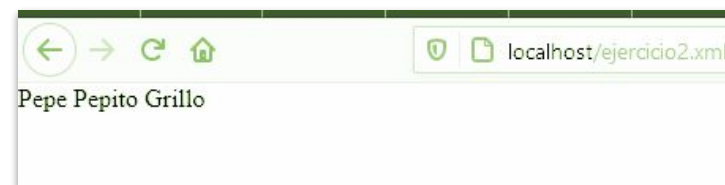
<!DOCTYPE identificacion [
  <!ELEMENT identificacion (nombre_completo, apodo)>
  <!ELEMENT nombre_completo (nombre, apellido1, apellido2)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT apellido1 (#PCDATA)>
  <!ELEMENT apellido2 (#PCDATA)>
  <!ELEMENT apodo (#PCDATA)>
]>
<identificacion>
  <nombre_completo>
    <nombre> Pepe </nombre>
    <apellido1> González </apellido1>
    <apellido2> Ribera </apellido2>
  </nombre_completo>
  <apodo> Pepito Grillo </apodo>
</identificacion>
```

XML con DTD y XSLT

```
<?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="nombre_completo">
    <xsl:value-of select="nombre"/>
  </xsl:template>
</xsl:stylesheet>
```

XSLT



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

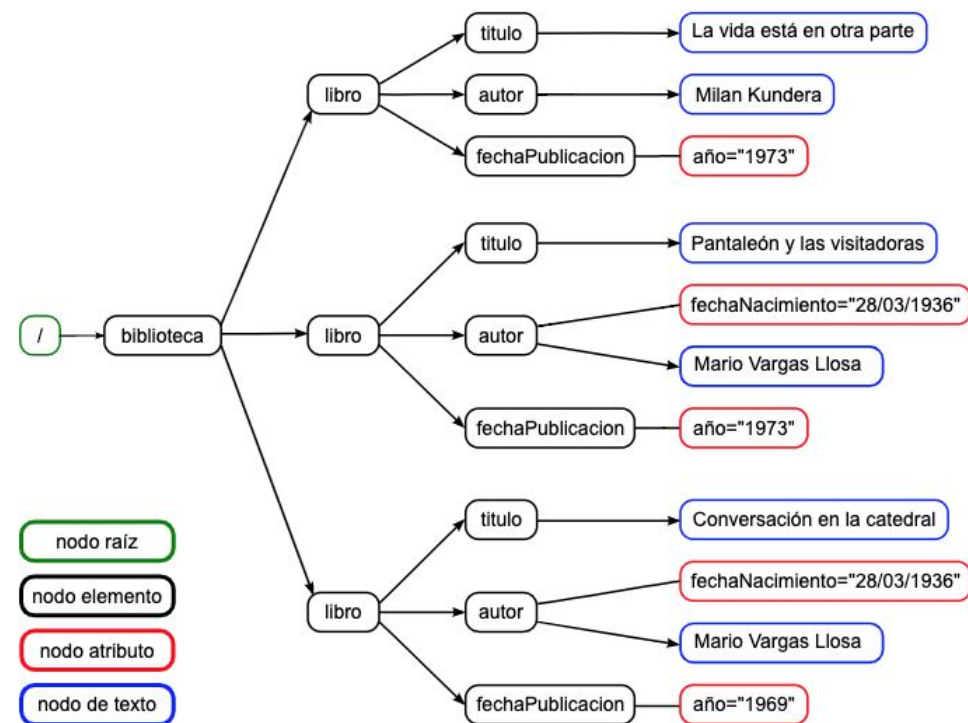
TRANSFORMACIÓN

4. Ejemplos de plantillas

```

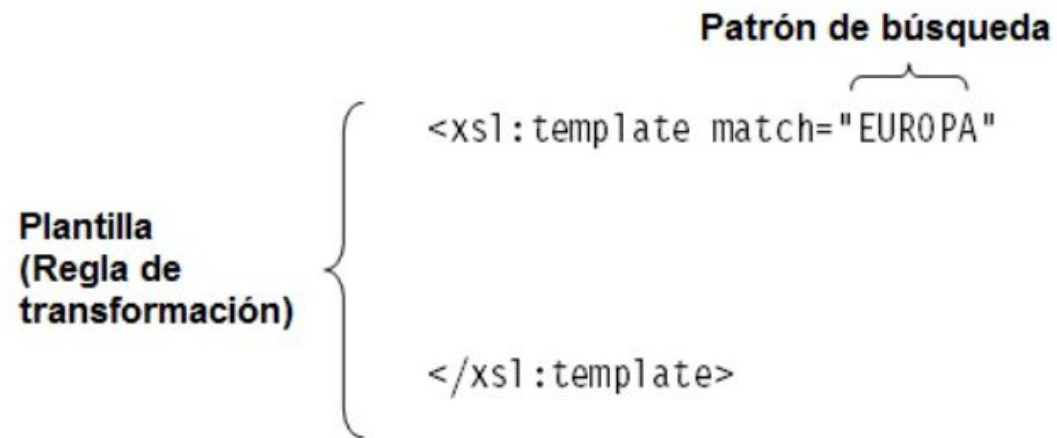
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas 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>

```



Plantillas

Las reglas de plantilla (template) son patrones (patterns) para la transformación del árbol fuente en el árbol resultado. Éstas se componen de dos partes, la primera se conoce como patrones de búsqueda y la segunda como plantillas. A través del patrón se identifican los nodos del árbol fuente a los cuales se aplica la regla.



Ejemplos de plantillas

Vamos a ver diferentes tipos de plantillas usando el siguiente XML:

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

CUANDO NO HAY PLANTILLA

```
<?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
```

Al no existir la plantilla, nos devuelve los datos de todos los nodos

Ejemplos de plantillas

Vamos a ver diferentes tipos de plantillas usando el siguiente XML:

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

```
<?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
```

En el ejemplo siguiente, el resultado incluye el contenido de los nodos <titulo>, ya que no hay regla para ellos, pero los de <autor> se pierden porque la plantilla es vacía

Ejemplos de plantillas

Vamos a ver diferentes tipos de plantillas usando el siguiente XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 VACÍA O NO EXISTENTE

```
<?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:stylesheet>
```

RESULTADO

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

En este caso si la plantilla vacía afecta a la raíz del XML, entonces devolverá un resultado totalmente vacío.

Ejemplos de plantillas

La instrucción <xsl:value-of>

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

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

La instrucción <xsl:value-of>

```
<?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

Ejemplos de plantillas

La instrucción <xsl:value-of>

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

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

La instrucción <xsl:value-of>

```
<?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
```

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.

Ejemplos de plantillas

La instrucción <xsl:value-of>

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

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

La instrucción <xsl:value-of>

```
<?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
```

En este caso, el ".", indica que va a sacar el contenido del propio elemento, el autor, además también muestra el resto del contenido por no haber una regla para el mismo.

Ejemplos de plantillas

La instrucción <xsl:value-of>

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

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

La instrucción <xsl:value-of>

```
<?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
```

En este caso, con @ se pueden sacar los atributos, de esta forma sacamos el año y el resto del contenido al no haber una regla para ellos

Ejemplos de plantillas

Generar texto adicional

Se puede generar texto escribiéndolo en la regla, por ejemplo, código html.

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

La instrucción <xsl:value-of>

```
<?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>
```

El resultado sigue sin verse bien en el navegador, porque aunque hay etiquetas <p>, falta la etiqueta global <html>.

Ejemplos de plantillas

Generar texto adicional

Se puede generar texto escribiéndolo en la regla, por ejemplo, código html.

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

La instrucción <xsl:value-of>

```
<?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>
```

El resultado sigue sin verse bien en el navegador, porque aunque hay etiquetas <p>, falta la etiqueta global <html>.

Ejemplos de plantillas

Generar texto adicional

Se puede generar texto escribiéndolo en la regla, por ejemplo, código html.

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

La instrucción <xsl:value-of>

```
<?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>
```

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).

Ejemplos de plantillas

Generar texto adicional

Se puede generar texto escribiéndolo en la regla, por ejemplo, código html.

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

La instrucción <xsl:value-of>

```
<?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>
```

Ejemplos de plantillas

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

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

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

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="/">
    <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.

Ejemplos de plantillas

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

- 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.

Ejemplos de plantillas

La instrucción <xsl:strip-space>

Aplicable en XML Copy Editor

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

La instrucción <xsl:strip-space>

```
<?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>
```

Ejemplos de plantillas

La instrucción <xsl:text>

Aplicable en Notepad++

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

La instrucción <xsl:strip-space>

```
<?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:text>&#10; </xsl:text>
      <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>
```

Ejemplos de plantillas

La instrucción <xsl:attribute>

La instrucción <xsl:attribute> permite generar un atributo y su valor.

Se utiliza cuando el valor del atributo se obtiene a su vez de algún nodo.

Vamos a ver el resultado en el siguiente XML:

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

La instrucción <xsl:attribute>

```
<?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>
```

En la hoja de estilo XSLT, la etiqueta se escribe con apertura y cierre, aunque en el resultado aparece como etiqueta monoatómica.

Ejemplos de plantillas

La instrucción <xsl:attribute>

Para que se vea en el navegador es necesario poner la etiqueta <html>

La instrucción <xsl:attribute>

```
<?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>
```

Enlaces de interés

1. [W3 Schools - DTD Tutorial](#)
2. [W3Schools XML Tutorial](#)
3. [XML Copy Editor](#)

Bibliografía

Todo el contenido del documento ha sido extraído de <https://www.mclibre.org/consultar/xml/lecciones/xml-xslt.html>

