

Tenemos el XML de biblioteca el cual está definido de la siguiente forma:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="ejemplol.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>
```

Preguntas a responder:

```
<?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"/>
</xsl:stylesheet>
```

Esto no transforma nada, lo único que hace es una estructura base para una transformación XSLT, pero no hace nada.

```
<?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"/>
  <xsl:template match="/">
  </xsl:template>
</xsl:stylesheet>
```

La diferencia con el anterior y este, es que aquí si hemos definido una plantilla en el nodo raíz "/". Pero sigue sin tener funcionamiento como tal.

```
<?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"/>

  <xsl:template match="/">
    Este texto se muestra
  </xsl:template>

  <xsl:template match="titulo">
    ¿Esto se muestra?
  </xsl:template>

</xsl:stylesheet>
```

Esta nos mostrará el primer texto, ya que definimos una ruta raíz que está en "/" y titulo no la contiene por lo tanto no se muestra.

```
<?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"/>

  <!--<xsl:template match="/">
    Este texto se muestra
  </xsl:template>-->

  <xsl:template match="titulo">
    ¿Esto se muestra?
  </xsl:template>

</xsl:stylesheet>
```

Ahora que hemos quitado la raíz, se nos mostrará el texto del match="titulo" y aparte de este texto se recorre el xml imprimiendo los títulos de los libros.

```
<?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"/>

  <!--<xsl:template match="/">
    Este texto se muestra
  </xsl:template>-->

  <xsl:template match="libro">
    Esto es un ejemplo para obtener el valor de un nodo.
    Por ejemplo mostramos los nodos autor.
    <xsl:value-of select="."/>
  </xsl:template>

</xsl:stylesheet>
```

Ahora con este caso lo que estamos sacando es, como antes el texto y aparte de esto

damos toda la información de cada objeto del xml. Es decir damos la información dentro de etiquetas pero no atributos.

```
<?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"/>

  <!--<xsl:template match="/">
    Este texto se muestra
  </xsl:template-->

  <xsl:template match="libro">
    Esto es un ejemplo para obtener el valor de un nodo.
    Por ejemplo mostramos los nodos autor.
    - <xsl:value-of select="autor"/>
    Además podemos mostrar el valor de un atributo
    - <xsl:value-of select="fechaPublicacion/@año"/>
  </xsl:template>

</xsl:stylesheet>
```

En este caso, la información que se nos muestra es obviamente la misma que antes pero agregando la fecha de publicación del libro. Esto pasa porque referenciamos al atributo en el segundo select.

Si quisiéramos que además saliera la fecha de nacimiento del autor, simplemente tendríamos que agregar esta línea:

```
<xsl:value-of select="autor/@fechaNacimiento"/>
```

```
<xsl:template match="autor">
  <p>Nombre:<xsl:value-of select="."/></p>
  <p>Fecha de nacimiento:<xsl:value-of select="@fechaNacimiento"/></p>
</xsl:template>
```

Esto nos dará tanto la info del autor (Nombre) como el atributo fecha Nacimiento de este mismo.

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

  <!--<xsl:template match="/">
    Este texto se muestra
  </xsl:template>-->

  <xsl:template match="libro">
    Esto es un ejemplo para obtener el valor de un nodo.
    Por ejemplo mostramos los nodos autor.
    - <xsl:value-of select="autor"/>
    Además podemos mostrar el valor de un atributo
    - <xsl:value-of select="fechaPublicacion/@año"/>
  </xsl:template>

</xsl:stylesheet>
```

En este caso nos dará el valor del autor y el atributo de fecha de publicación pero en formato texto y no HTML , es decir. No podríamos usar etiquetas propias de este.

```
<?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"/>

  <xsl:template match="/">
    Este texto se muestra
    <xsl:apply-templates />
  </xsl:template>

  <xsl:template match="libro">
    Esto es un ejemplo para obtener el valor de un nodo.
    Por ejemplo mostramos los nodos autor.
    - <xsl:value-of select="autor"/>
    Además podemos mostrar el valor de un atributo
    - <xsl:value-of select="fechaPublicacion/@año"/>
  </xsl:template>

</xsl:stylesheet>
```

Ahora con este ejemplo hemos cambiado al caso anterior, hemos agregado la línea `xsl:apply-templates`. Que lo que hace es que no se limita a un único template sino que devuelve los datos de más de uno, así que nos imprimirá el texto del "/" y los datos del "libro".


```
<?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"/>

  <xsl:template match="/">
    Este texto se muestra
    <xsl:apply-templates />
  </xsl:template>

  <!--<xsl:template match="libro">
    Esto es un ejemplo para obtener el valor de un nodo.
    Por ejemplo mostramos los nodos autor.
    - <xsl:value-of select="autor"/>
    Además podemos mostrar el valor de un atributo
    - <xsl:value-of select="fechaPublicacion/@año"/>
  </xsl:template-->

  <xsl:template match="autor">
    <p>Nombre:<xsl:value-of select="."/> </p>
    <p>Fecha de Nacimiento: <xsl:value-of select="@fechaNacimiento"/> </p>
  </xsl:template>
```

Con esta estructura nos dará ambos templates y aparte se nos mostrará la información del autor, tanto el nombre de este como su fecha de nacimiento.

```
<xsl:template match="/">
  Este texto se muestra
  <xsl:apply-templates select="autor"/>
</xsl:template>
```

Si hacemos esto , lo que pasará es que solo se mostrará el mensaje, ya que se referencia 2 veces a lo mismo y es error lógico.

```
<?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"/>

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

  </xsl:stylesheet>
```

En este caso como coincide en match con el atributo seleccionado (Están en el mismo nivel) La información se imprime, antes no coinciden la raíz y donde queríamos acceder.

