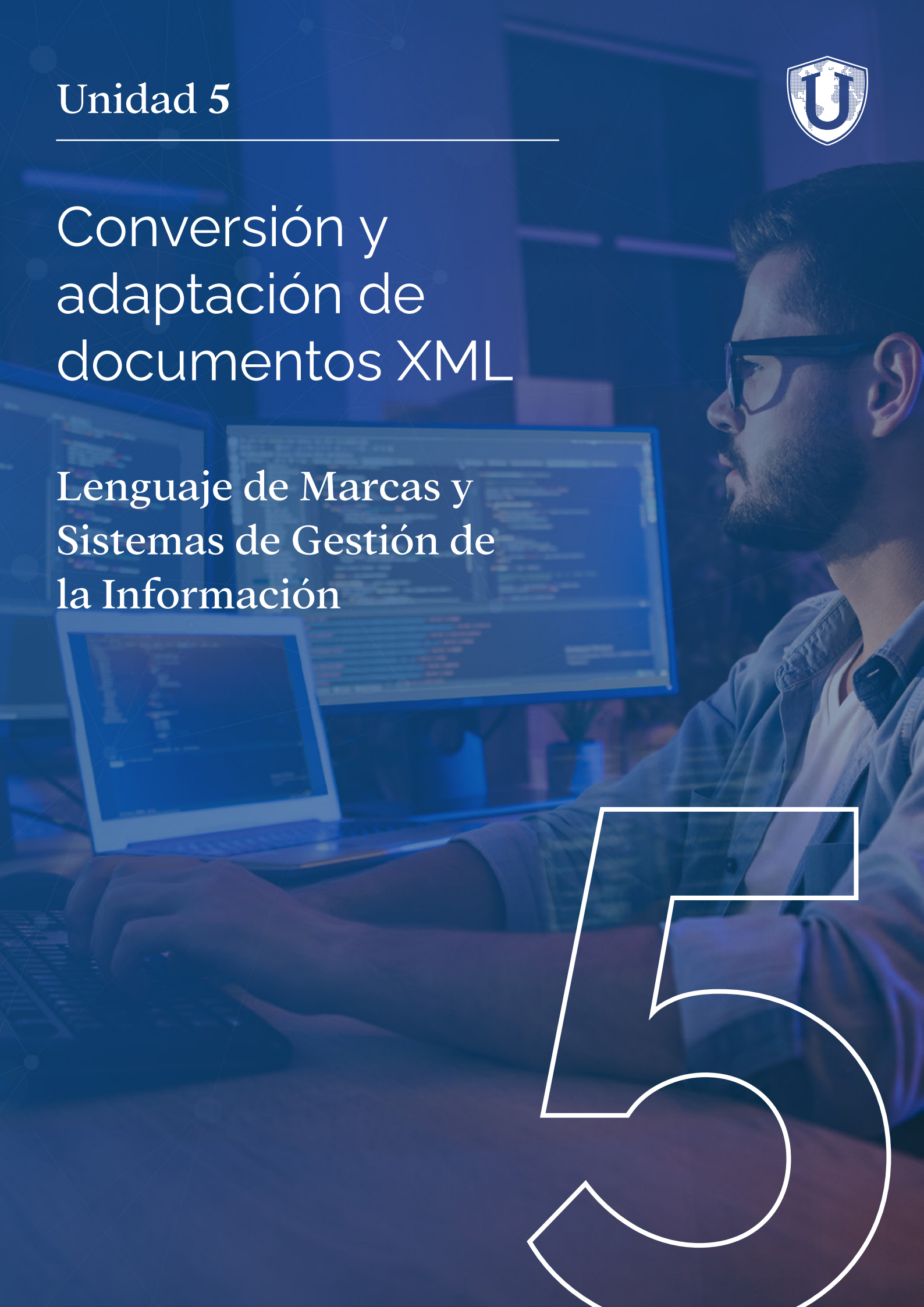
Unidad 5

Conversión y adaptación de documentos XML

Lenguaje de Marcas y Sistemas de Gestión de la Información



5

Índice

Introducción

En este tema descubriremos como transformar los archivos XML con archivos XSL, extensible stylesheet language, de modo que podamos llevar al mundo web, en su forma final, los archivos XML que hemos creado.

Para lograr este objetivo veremos las distintas versiones de las hojas de estilo XSL y las líneas más importantes de estas junto con las etiquetas imprescindibles que debemos conocer para la creación de nuestras propias hojas de estilo.

Por último, veremos las diferentes herramientas que un desarrollador puede usar para facilitar su trabajo junto a los diferentes formas y programas que tenemos a nuestra disposición para adquirir dichas herramientas.

Al finalizar esta unidad

- Aprenderemos a transformar documentos XML con XSLT y las herramientas XPath a otros formatos, como es el HTML.
- Distinguiremos las diferencias entre las distintas versiones del XSLT.
- Descubriremos la sintaxis y estructura de los archivos XSL, así como la manera de transformar archivos XML en HTML.

5.1. Técnicas de transformación de documentos XML

Las transformaciones XSLT permiten la modificación de elementos, atributos, etc. así como modificar la estructura con el fin de reestructurar y ordenar los elementos del sistema.

También permite la unión de dos archivos, un árbol de origen con una hoja de estilo con el fin de crear un árbol de destino.

La hoja de estilo, o style sheets, permite añadir al documento XML, el árbol de origen, una serie de características y reglas a la hora de visualizarlo, como pueden ser:

- El color.
- El fondo.
- Tipo de fuente.
- Apariencia de los bordes.
- Márgenes.
- Alineación y espacio entre caracteres.

El árbol de destino incluirá todas estas normas de estilo incluidas en su código. Las hojas de estilo más usadas son XSL y CSS, aunque la transformación XSLT aceptará archivos XML y XSL, pero no CSS, lo que dará como resultado un archivo de salida.

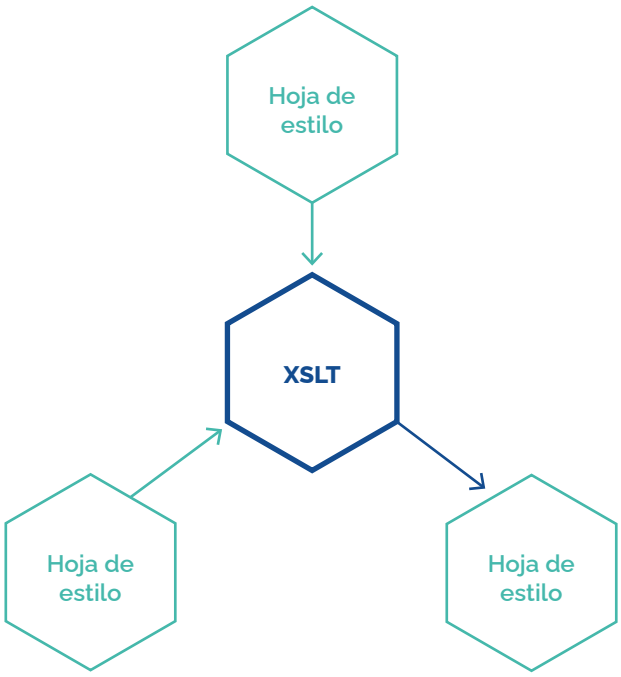


Imagen 1. Proceso de la transformación XSLT.

Puedes encontrar el software XSLT en editores XML o en el propio navegador.

5.1.1. Diferencias entre las versiones de XSLT 1.0, 2.0 y 3.0

La primera versión de XSLT surgió en 1999 y hasta 2007 no surgió una nueva versión, en esta época se creó en XSLT 2.0 que mejoró sus prestaciones, aunque seguía siendo compatible con versiones anteriores, es decir, era retrocompatible. Los XSLT más usados son Xalan, Saxon y MSXML.

La versión 2.0 cambio en los siguientes aspectos:

- Mejora del sistema de tipos con una gran ampliación de su oferta.
- Mejora de las funciones.
- Mejora del modelo de datos.
- Introducción de soporte para expresiones regulares.
- Posibilidad de crear múltiples arboles simultáneamente.
- Uso de la etiqueta <xsl:function>. Para expresiones XPath.
- Mayor calidad del sistema de errores.
- Mejora en la agrupación.

Se crea en esta versión 2.0 la posibilidad de streaming de documentos con el fin de trabajar con el documento aún incompleto, para ello se crearon las siguientes instrucciones: <xsl:source-document>, <xsl:iterate>, <xsl:merge> o <xsl:fork>.

También se añadió el concepto de paquete, con el que poder trabajar sobrescribiendo datos, aunque ya los estemos visualizando, lo cual favorece el mantenimiento del código.

De entre las expresiones añadidas destacan las siguientes:

- **xsl:evaluate.** Evalua expresiones XPath constituidas por strings.
- **xsl:try.** Que evita el bloqueo a causa de errores.
- **xsl:global-context-item.** Permite especificar un elemento en un contexto global.
- **xsl:assert.** Permite construir un código más robusto libre de errores paralizantes.

Existe una versión 3.0, pero la más actual es la segunda edición de la versión 2.0 marzo de 2021.

XPath es un lenguaje que permite seleccionar y navegar entre los distintos nodos de XML.

5.2. Formatos de salida y ámbitos de aplicación

La transformación XSLT da como resultado un documento de salida, el cual puede ser de varios formatos, como pueden ser el XML y el XHTML.

5.3. Descripción de la estructura y la sintaxis

5.3.1. Fichero XML

La asociación del XSL al XML se realizará mediante la siguiente línea:

```
<?xml-stylesheet type="text/xsl" href="XYZ.xsl"?>
```

La cual se escribirá bajo la primera línea común a ambos archivos, donde se indica el tipo de archivo, versión y encoding: <?xml version="1.0" encoding="UTF-8"?> manteniendo la misma línea en ambas. En la anterior el lugar que ocupa "XYZ.xsl" en el atributo "href" es el que expresará el nombre del archivo XSL, no se debe olvidar que es necesario escribir también la extensión.

Otra línea imprescindible es: <xsl:output method="html"/>, la cual marca el formato con el que acabará el archivo tras la transformación XSLT, en este caso el archivo terminará con un formato HTML.

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

<?xml-stylesheet type="text/xsl" href="Formato
de estilo.xsl"?>

<xsl:output method="html"/>

<inventario>

        <plantas>

                <especie>Rosa</especie>

                <precio>1 €</precio>

        </plantas>

        <plantas>

                <especie>clavel</especie>

                <precio>0.75 €</precio>

        </plantas>

</inventario>
```

1. Tipo de documento XML.
2. Referencia del documento XSL.
3. Indica el formato de salida en una transformación XSLT.

Imagen 2. Ejemplo XML asociado a XSL

5.3.2. Hoja de estilo XSL

Los archivos XSL deben comenzar con la misma línea que los archivos XML

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

Pero a diferencia de ellos deberán contener una segunda línea que los marcará como archivos XSL:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.
w3.org/1999/XSL/Transform">
```

La siguiente parte será la línea <xsl:template match="/">, la cual, con el atributo "match" indica a que elementos del archivo XML se asociará el formato del XSL, en este caso la expresión XPath "/" indica que se asocia a toda la raíz y, por ende, a todo el documento.

Expresiones XPath

Expresión	Elementos seleccionados
/	La raíz y, por tanto, todo el documento
/Objeto	Solo el especificado, en este caso "Objeto"
//Objeto	Todos los "Objeto" del documento
/Objeto/*/Objeto2	El especificado "Objeto2" que tienen como padre "Objeto"

5.4. Utilización de plantillas

La creación de plantillas, para archivos XML es sencilla, gracias a las etiquetas específicas que se verán a continuación y que reducen su dificultad significativamente.

5.4.1. <xsl:value-of>

La etiqueta <xsl:value-of> extraen el valor del elemento referenciado en ella desde el archivo XML.

```
<xsl:value-of select=”especie/precio”/>
```

Esta línea extraería los valores de “especie” y “precio” especificados en el documento XML. Su único uso es la selección.

5.4.2. <xsl:for-each>

Esta etiqueta selecciona todos los elementos especificados, en el próximo ejemplo serán “especie” y “precio”, en el conjunto de nodos que abarquen.

```
<xsl:for-each select=”especie/precio”>
</xsl:for-each>
```

Es posible acotar la selección con una instrucción añadida:

```
<xsl:for-each select=”especie/precio[flores=’Rosa’]”>
</xsl:for-each>
```

Que limitará la búsqueda según el campo seleccionado y el filtro usado:

Códigos de filtros

Filtros	Significado
=	Igual
!=	Desigual
<	Menos que
>	Mayor que

5.4.3. Filtrando contenido

Gracia a los códigos de filtrado de la etiqueta <xsl:for-each> podemos filtrar el documento, de modo que podamos trabajar con una parte del archivo, lo que facilita su elaboración.

Podéis intentarlo en la página w3schools.com:

https://www.w3schools.com/xml/tryxslt.asp?xmlfile=cdcatalog&xsltfile=cdcatalog_filter.

5.4.4. Ordenando el contenido

Otras etiquetas como <xsl:sort> la cual permite ordenar el contenido según el valor añadido al atributo “select”, en el caso de tratarse de números se ordenará desde menor a mayor, y si se tratase de nombres se agruparían los campos repetidos.

```
<xsl:sort select=”Precio”/>
```

5.4.5. Selección avanzada IF

Para un filtrado de mayor precisión podemos emplear la etiqueta "<xsl:if>", la cual, con el atributo "test" y los caracteres numéricos nos permite delimitar los filtrados. Siempre debe encontrarse dentro de la etiqueta "<xsl:for-each>".

```
<xsl:for-each select="especie/precio">
        <xsl:if test="precio<1">
                <tr>
                        <td><xs1:value-of select="especie"/></td>
                        <td><xs1:value-of select="precio"/></td>
                </tr>
        </xsl:if>
</xsl:for-each>
```

5.4.6. Selección avanzada CHOOSE

La etiqueta <xsl:choose> permite seleccionar un conjunto de elementos, delimitados en la etiqueta que lo acompaña, <xsl:when test="AAA>1 "> </xsl:when>, donde el atributo test señala un atributo que se delimitará de la misma manera que la etiqueta <xsl:if>. Esta delimitación permite realizar ciertas acciones con esas celdas delimitadas, como por ejemplo colorearlas de cierto color, de modo que podemos ver, por ejemplo, si un valor supera cierto límite un cambio de color que los destaque.

Podemos incluir la etiqueta <xsl:otherwise> </xsl:otherwise> para delimitar el resto de campos no delimitados por la etiqueta <xsl:when>.

```
<xsl:choose>
        <xsl:when test="precio>10">
        …
        </xsl:when>
        <xsl:otherwise>
        …
        </xsl:otherwise>
</xsl:choose>
```

De modo que con esta estructura se señalarán, por un lado, los elementos con un precio superior a 10 y, por otro, los elementos con un precio inferior a 10.

5.5. Utilización de herramientas de procesamiento: verificación, depuración y generación de documentación

Con el fin de facilitar el trabajo con los archivos XML se han creado una multitud de herramientas, tanto de pago como de código abierto, open source.

Una de las herramientas más utilizadas son los depuradores, los cuales permiten revisar los documentos en busca de errores, al permitir visualizar la transformación de los códigos XML, facilitando en gran medida la tarea. Estos elementos no suelen estar en las herramientas de libre código, pero si en los de pago.

5.5.1. Oxygen XML Editor

Oxygen XML Editor es uno de los editores XML disponibles más usados, ya que proporciona una amplia gama de herramientas de creación y desarrollo XML, en múltiples plataformas. Su interfaz permite su uso tanto por desarrolladores expertos como novatos. Está diseñado para su uso en una multitud de plataformas y contiene las herramientas imprescindibles para cualquier desarrollador.

Algunas de sus características destacadas son:

- Edición XML estructurada.
- Compatible con una multitud de plataformas.
- Herramientas de validación.
- Depuración XSLT.

5.5.2. Altova Editor XSLT

Altova Editor XSLT es un editor con una gran cantidad de herramientas que permiten al desarrollador realizar su trabajo con todas las facilidades posibles, además cuenta con herramientas avanzadas como un evaluador de XPath, explorador web integrado y transformación de alta velocidad. Estos últimos pueden realizar cambios en el documento con el fin de que la transformación se realice con mayor celeridad, ya que en documentos especialmente largos este proceso puede conllevar un tiempo considerable.

Altova, entre otras cosas nos permitirá:

- La generación automática hojas de estilo XSLT 1.0/2.0/3.0 y XSL:FO.
- La creación un diseño a partir de un archivo XSLT existente o importar código XSLT externos.
- El diseño de formularios electrónicos interactivos para actualizar datos XML y de bases de datos.
- Generar aplicaciones web ASPX a partir de diseños.

5.5.3. Saxon

La empresa Saxonica Limited desarrollo su propio procesador XSLT, el Saxon, el cual permite a los desarrolladores trabajar en múltiples plataformas, con varios lenguajes. Cuenta con tres versiones:

- **Home Edition (HE)**: Edición de código abierto, gratuito.
- **Professional Edition (PE)**: Edición con características adicionales para profesionales.
- **Enterprise Edition (EE)**: Edición con optimizaciones para un alto rendimiento y productividad.

5.5.4. XPontus XML Editor

XPontus XML Editor es un sencillo editor XML de código abierto, que se presenta como una alternativa gratuita a los programas de pago.

Algunas de sus características son:

- Poder realizar validación.
- Realizar transformaciones XSL.
- Generación de esquemas.

5.5.5. Herramientas online

Una alternativa a los programas mencionados son las herramientas online. Estas herramientas poseen muchas de las características de los programas de escritorios, pero con la comodidad que ofrecen las herramientas online. Uno de estos programas online recomendados es XSL Test Online.

Como desventaja a este tipo de programa online se debe mencionar que para una programación profesional puede no resultar ideal, ya que los programas online suelen constar con menos prestaciones y en especial carecen de depuradores que, para archivos de gran envergadura, son imprescindibles.

www.universae.com