Tarea 5: Conversión y Adaptación de Documentos XML

Francisco Javier Sueza Rodríguez

28 de abril de 2023

Centro:	IES Aguadulce
Ciclo Formativo:	Desarrollo Aplicaciones Web (Distancia)
Asignatura:	Lenguajes de Marcas y Sistemas de Gestión de la Información
Tema:	Tema 5 - Conversión y Adaptación de Documentos XML

1. Caso Práctico

La empresa de reparto LlegaYA, S.L. sigue avanzando. Es momento de utilizar toda nuestra información recogida en formato XML. Realizaremos la conversión y adaptación de documentos XML a otros formatos.

Todo esto nos facilitará exportar nuestra información a la web. Además podremos realizar búsquedas y filtrar la información con los criterios que necesitemos.

2. Actividades

2.1. Enunciado

En la empresa LlegaYA, S.L. queremos automatizar unos informes mensuales. Nuestro sistema informático registra mensualmente las entregas realizadas que recogemos en el archivo: envio.xml.

Partiendo de la plantilla: envio.xsl, queremos que elabores las siguientes consultas en el formato establecido en cada una.

A. Lista ordenada por precio y apellido de los envíos a Sevilla. Indicar el número de orden (con número), el precio, la moneda, el apellido y el nombre. El orden será de mayor a menor precio y si tienen el mismo precio por orden alfabético de apellido.

   Formato:

   1) 33 euros - Sánchez, Carlos.

B. Número de envíos urgentes a Cádiz y su porcentaje respecto al total de envíos a Cádiz

   Formato:

   Hay 4 envíos urgentes a Cádiz, que suponen el 28.57 % de los 14 envíos totales registrados a Cádiz.

C. Lista ordenada (por código de envío) con el tipo de prioridad, la provincia, el nombre y el apellido de todos los envío cuyo nombre comience por 'A' y tengan una prioridad 'Normal', o su apellido contenga una 'a' y la provincia sea 'Almería' o 'Granada'.

   Formato:

   1.- (DBD72R - 24_horas - Granada). Carlos Cano.

D. Lista de todas las provincias (ordenadas alfabeticamente) con su número de envíos, ingresos totales (suma de todos sus precios) e ingreso medio.

   Formato:

   Almería: 11 envíos. Ingresos totales: 229 euros. Ingreso medio: 20.82 euros.

E. Crear una tabla, ordenada por fecha de entrega, de los envíos a Almería. La tabla incluirá las columnas: fecha de entrega, provincia, código de envío y prioridad.

   Estilos:

   La tabla deberá usar los estilos definidos en la plantilla que se proporciona en el ejercicio. Los elementos tabla y las celdas usarán los estilos del descriptores 'table', 'th' y 'td'. La cabecera usará el estilo del descriptor 'th'. Si la prioridad de un envío es 'Urgente' esa celda usará el estilo

del descriptor '.urgente'. Si la prioridad de un envío es 'Nocturno' esa celda usará el estilo del descriptor '.nocturno'.

Formato:

Fecha	Provincia	Código de envío	Prioridad
Formato de tabla requerido.		??????	Normal
2023-02-??	Almería	??????	Urgente
2023-02-??	Almería	??????	Nocturno

Figura 2.1: Tabla de Envíos

2.2. Solución

En este ejercicios vamos a realizar diversas transformaciones al documento XML de ejemplo para que se nos muestre la información que se nos pide en cada punto. Se incluirá solo el código agregado en cada punto, no la salida.

A. Para resolver este punto, se ha creado un bucle, con ***<xsl:for-each>***, donde se han incluido, en primer lugar, dos instrucciones ***<xsl:sort>***, una para ordenar por el precio de forma descendente, y otro para ordenar por el apellido, de forma ascendente, por defecto.

   A continuación se ha creado un elemento ***xs:element*** de tipo ***<p>*** por cada nodo en el que se ha incluido la información pedida, la cual se ha obtenido con la instrucción ***xsl:value-of***, realizando el ***select*** según la información que se nos pedía, es decir ***postition()***, ***precio*** , ***apellidos*** y ***nombre***.

```
<xsl:for-each select="//envio[provincia='Sevilla']">
  <xsl:sort select="precio" data-type="number" order="descending" />
  <xsl:sort select="apellido" />
  <p>
    <xsl:value-of select="position()" />)
    <xsl:value-of select="precio" /> euros -
    <xsl:value-of select="apellido"/>,
    <xsl:value-of select="nombre" />
  </p>
</xsl:for-each>
```

B. Para resolver este ejercicios se han usado, en primer lugar, **3 variables**. Una para almacenar el **nombre de la provincia**, otra para el número de **envios totales** y otra para los **envíos urgentes**. La variable ***$prov***, que almacena el nombre de la provincia, se han empleado en la selección de nodos tanto de ***$env_totales*** como de ***$env_urg***, de esta forma, si quisiéramos mostrar la información de alguna otra provincia, por ejemplo Sevilla, solo habría que cambiar el valor de la variable ***$prov*** y se mostrarían los datos correctamente. Además, se ha empleado la función ***count()*** para el cálculo de la cantidad de envíos en las respectivas variables.

   A continuación se ha creado un elemento de tipo ***<p>*** mediante ***xsl:element*** y dentro se han usado diferentes instrucciones de tipo ***xsl:value-of*** para mostrar la información requerida. Cabe mencionar el caso del **cálculo del porcentaje**, donde se ha usado en el ***select*** la función ***format-number*** para dar un formato adecuado a la salida del cálculo del porcentaje, ya que por defecto muestra demasiados decimales.

```
<xsl:variable name="prov" data-type="text" select="'Cádiz'" />
<xsl:variable name="env_urg" select="count(//envio[provincia=$prov and prioridad='Urgente'])" />
<xsl:variable name="env_total" select="count(//envio[provincia=$prov])" />

<xsl:element name="p">
  Hay <xsl:value-of select="$env_urg" /> envíos urgentes a <xsl:value-of select="$prov" />,
  que supone el  <xsl:value-of select="format-number(($env_urg div $env_total) * 100, '#.##')" />%
  del los <xsl:value-of select="$env_total" /> envíos totales registrados a
  <xsl:value-of select="$prov" />.
</xsl:element>
```

C. Para realizar esta consulta se han usado varias operaciones lógicas en el select del **for-each** principal. Por un lado comprobamos que el nombre de los envíos empiece por 'A' y tenga prioridad Normal ó que el nombre contenga la vocal 'a' y que la provincia sea Granada o Almería.

Una vez seleccionados los nodos adecuados en el for-each, simplemente se ordenan por el número de código, con xsl:sor y se muestra la información dentro de un elemento <p>creado con xsl:element.

```
<xsl:for-each select="
  //envio[(starts-with(nombre, 'A') and prioridad='Normal') or
  (contains(nombre, 'a') and (provincia='Granada' or provincia='Almería'))]">

  <xsl:sort select="@codigo" />
  <xsl:element name="p">
    <xsl:value-of select="position()" />.-
    (<xsl:value-of select="@codigo" /> -
    <xsl:value-of select="prioridad" /> -
    <xsl:value-of select="provincia" />).
    <xsl:value-of select="nombre" />
    <xsl:text> </xsl:text>
    <xsl:value-of select="apellido" />.
  </xsl:element>

</xsl:for-each>
```

D. Para resolver esta consulta en primer lugar hemos usado un **for-each** para seleccionar los envios con un predicado que incluye **provincia=preceding::provincia**, para que vaya seleccionando solo las provincias con el mismo nombre en cada pasada.

A continuación, se han creado 4 variables, **prov** que contiene el nombre de la provincia, **num_envios**, que usa count() para obtener el número total de envíos de esa provincia, **ingresos**, donde se ha usado sum() para calcular el total de ingresos y por último **media**, donde se han usados las 2 variables anteriores para calcular los ingresos medios.

```
<xsl:for-each select="//envio[not(provincia=preceding::provincia)]">
  <xsl:sort select="provincia" order="ascending" data-type="text" />

  <xsl:variable name="prov" select="provincia" />
  <xsl:variable name="num_envios" select="count(//envio[provincia=$prov])" />
  <xsl:variable name="ingresos" select="sum(//envio[provincia=$prov]/precio)" />
  <xsl:variable name="media" select="format-number($ingresos div $num_envios, '#.##')" />

  <xsl:element name="p">
    <xsl:value-of select="$prov" />: <xsl:value-of select="$num_envios" />.
    Ingresos totales: <xsl:value-of select="$ingresos" />.
    Ingresos Medios: <xsl:value-of select="$media" /> euros.
  </xsl:element>
</xsl:for-each>
```

E. En este caso se ha creado una tabla, empleando las **etiquetas html**. En la parte del cuerpo de la tabla, se ha usado un **xsl:for-each** para iterar por todos los envios en la provincia de Almería, creado una fila mediante **xsl:element** para cada uno.

   Se han rellenado cada celda usando **xsl:value-of**, con los diferentes datos. Cabe mencionar que en la última celda, referente a la **prioridad**, se ha creado ésta con **xsl:element**, y se ha usado **xsl:attribute**, creando un atributo '**class**' y usando **xsl:choose** para cargar una u otra clase CSS dependiendo del tipo de priodidad.

```
<table>
  <thead>
    <tr>
      <th>Fecha</th>
      <th>Provincia</th>
      <th>Código de envío</th>
      <th>Prioridad</th>
    </tr>
  </thead>
  <tbody>
    <xsl:for-each select="//envio[provincia='Almería']" >

      <xsl:element name="tr">
        <td><xsl:value-of select="fecha_entrega" /></td>
        <td><xsl:value-of select="provincia" /></td>
        <td><xsl:value-of select="@codigo" /></td>
        <xsl:element name="td">
          <xsl:attribute name="class">
            <xsl:choose>
              <xsl:when test="prioridad='Urgente'">urgente</xsl:when>
              <xsl:when test="prioridad='Nocturno'">nocturno</xsl:when>
            </xsl:choose>
          </xsl:attribute>
          <xsl:value-of select="prioridad" />
        </xsl:element>
      </xsl:element>

    </xsl:for-each>
  </tbody>
</table>
```