

INSTITUTO POLITÉCNICO NACIONAL

**UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERIA Y CIENCIAS SOCIALES Y ADMINISTRATIVAS**

**INTELIGENCIA DE NEGOCIOS**

**BIBLIOTECA – CLASIFICADOR**

* **OROZCO DÁVILA IGNACIO PABLO**
* **RIVERA DIAZ YASMIN CITLALLI**

**INTRODUCCIÓN**

El presente proyecto se enfocará principalmente a tareas de selección y ordenación de los datos. Los aspectos de representación, como la disposición tabular de los datos, se han dejado a las capacidades de HTML con instrucciones de estilo asociadas por medio de su atributo style.

Se han resaltado determinadas palabras de los títulos, que el número de edición se ha incluido entre paréntesis detrás de aquellos en la forma "(2a. ed.)", que a continuación se ha incluido una indicación de la novedad de la obra a base de asteriscos, y que cuando hay más de un autor se ha incluido "et al." en cursiva detrás del primero; todas estas decisiones se han tomado haciendo uso de posibilidades de XSLT.

**Elementos y restricciones**

El elemento *<xsl:value-of>*

Se usa para seleccionar el valor de un elemento XML y añadirlo al documento resultado de la transformación. El elemento se selecciona por medio del atributo obligatorio select, que toma como valor una expresión de XPath. En el ejemplo,

* select="$hoy - 5" resta 5 a la variable hoy (que fue definida por medio de <xsl:variable name="hoy">2005</xsl:variable>), dando como resultado "2000";
* select="edicion/@numedic" incorpora al documento de salida el valor del atributo numedic del elemento <edicion> (hijo del nodo actual, que es "/catalogo/libro" según se estableció con <xsl:for-each>);
* select="autor[1]" extrae el contenido del primer elemento <autor> que sea hijo del nodo actual;
* select="." extrae el contenido del nodo actual (que es el nodo <enfasis>).

El elemento *<xsl:for-each>*

Permite realizar bucles en XSLT. Puede usarse para seleccionar secuencialmente cada elemento XML de un determinado conjunto de nodos. El valor del atributo requerido select contiene una localización XPath.

En el ejemplo se ha usado, además de para seleccionar los elementos adecuados, esto es, cada <libro> del <catalogo>, con select="catalogo/libro" (la ruta es relativa al nodo actual, la raíz del documento tal como se estableció con el atributo match="/" de la plantilla en uso), también para filtrar determinados elementos, con el predicado añadido entre corchetes, select="catalogo/libro[edicion/anyo &gt;= $hoy - 5]", que contiene una expresión XPath relacional para seleccionar sólo aquellos libros cuyo año de publicación (<anyo>, hijo del elemento <edicion>) sea mayor o igual que 2000 (2005, valor de la variable hoy, menos 5).

El elemento <xsl:for-each select="child::\*">

Se usa cuando se busca las etiquetas hijos del XML, para el fin de identificar las etiquetas seleccionadas con el fin de buscar y seleccionar los datos de ello, osea que aplica en una plantilla a cada nodo, que en este caso, se busca los nodos principales que usa la biblioteca.

**ETL**

Es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, data mart, o ***data warehouse (DW)*** para analizar, o en otro sistema operacional.

En este caso, se realiza el ETL usando el XSLT para clasificar los datos de la biblioteca, incluyendo, determinar los datos por sus atributos, que en esta ocasión se usa *texto (text) y entero (int)*. Para llevarse a cabo en usar los *queries* en SQL para cargarlos en el DataWarehouse.

Gracias a ello, se puede crear el Data mart ya sea porque menores cantidades de datos implican que se procesan antes, tanto las cargas de datos como las consultas y se adopta a la arquitectura Top-Down.

**XML**

* **BD**

<?xml version="1.0" encoding="ISO-8859-1"?>

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

<catalogo>

<operaciones>

<disenio>

<libro>

<autor>texto</autor>

<titulo>texto</titulo>

<numedicion>entero</numedicion>

<anyo>entero</anyo>

</libro>

</disenio>

<datos>

<libro>

<autor>Matteo Pagani.</autor>

<titulo> Windows Phone 8 <enfasis> Development </enfasis> </titulo>

<numedicion>3</numedicion>

<edicion numedic="3">

<anyo>2002</anyo>

</edicion>

</libro>

<libro>

<autor>Chris Rose.</autor>

<titulo> Direct 3D </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>2001</anyo>

</edicion>

</libro>

<libro>

<autor>Stacia Misner.</autor>

<titulo> SQL Server Analysis Services </titulo>

<numedicion>4</numedicion>

<edicion numedic="4">

<anyo>2000</anyo>

</edicion>

</libro>

<libro>

<autor>Chris Rose.</autor>

<titulo> Assembly Language </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>2005</anyo>

</edicion>

</libro>

<libro>

<autor>Emanuele DelBono.</autor>

<titulo> <enfasis>ASP.NET</enfasis> Web API </titulo>

<numedicion>3</numedicion>

<edicion numedic="3">

<anyo>2002</anyo>

</edicion>

</libro>

<libro>

<autor>Joe Booth.</autor>

<titulo> Visual Studio Add-Ins </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>2004</anyo>

</edicion>

</libro>

<libro>

<autor>Peter Shaw.</autor>

<titulo> Postgres </titulo>

<numedicion>1</numedicion>

<edicion numedic="1">

<anyo>1999</anyo>

</edicion>

</libro>

<libro>

<autor>Chris Rose.</autor>

<titulo> Direct2D </titulo>

<numedicion>1</numedicion>

<edicion numedic="1">

<anyo>2001</anyo>

</edicion>

</libro>

<libro>

<autor>Stacia Misner.</autor>

<titulo> Windows Azure SQL Reporting </titulo>

<numedicion>3</numedicion>

<edicion numedic="3">

<anyo>2006</anyo>

</edicion>

</libro>

<libro>

<autor>Marc Clifton.</autor>

<titulo> Unit Testing </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>2003</anyo>

</edicion>

</libro>

<libro>

<autor>Steve Fenton.</autor>

<titulo> TypeScript </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>2004</anyo>

</edicion>

</libro>

<libro>

<autor>Buddy James.</autor>

<titulo> WPF </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>2006</anyo>

</edicion>

</libro>

<libro>

<autor>Ryan Hodson.</autor>

<titulo> iOS </titulo>

<numedicion>3</numedicion>

<edicion numedic="3">

<anyo>2005</anyo>

</edicion>

</libro>

<libro>

<autor>John Garland.</autor>

<titulo> Windows Store Apps </titulo>

<numedicion>1</numedicion>

<edicion numedic="1">

<anyo>2005</anyo>

</edicion>

</libro>

<libro>

<autor>Robert Horvick.</autor>

<titulo> Data Structures Part 1 </titulo>

<numedicion>1</numedicion>

<edicion numedic="1">

<anyo>2002</anyo>

</edicion>

</libro>

<libro>

<autor>Peter Shaw.</autor>

<titulo> GIS </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>1998</anyo>

</edicion>

</libro>

<libro>

<autor>Robert Horvick.</autor>

<titulo> Data Structures Part 2. </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>2003</anyo>

</edicion>

</libro>

<libro>

<autor>Ryan Hodson.</autor>

<titulo> Objective-C </titulo>

<numedicion>3</numedicion>

<edicion numedic="3">

<anyo>2004</anyo>

</edicion>

</libro>

<libro>

<autor>Robert Pickering.</autor>

<titulo> F# </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>1999</anyo>

</edicion>

</libro>

<libro>

<autor>Ryan Hodson.</autor>

<titulo> Knockout.js </titulo>

<numedicion>1</numedicion>

<edicion numedic="1">

<anyo>2005</anyo>

</edicion>

</libro>

<libro>

<autor>Michael McLaughlin.</autor>

<titulo> <enfasis>C++</enfasis> </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>2004</anyo>

</edicion>

</libro>

<libro>

<autor>Lyle Luppes.</autor>

<titulo> ASP.NET MVC 4 Mobile Websites </titulo>

<numedicion>1</numedicion>

<edicion numedic="1">

<anyo>2005</anyo>

</edicion>

</libro>

<libro>

<autor>Jan Van der Haegen.</autor>

<titulo> LightSwitch </titulo>

<numedicion>2</numedicion>

<edicion numedic="2">

<anyo>1998</anyo>

</edicion>

</libro>

<libro>

<autor>Cody Lindley.</autor>

<titulo> JavaScript </titulo>

<numedicion>3</numedicion>

<edicion numedic="3">

<anyo>2007</anyo>

</edicion>

</libro>

<libro>

<autor>Scott Allen.</autor>

<titulo> <enfasis>HTTP</enfasis> </titulo>

<numedicion>5</numedicion>

<edicion numedic="5">

<anyo>2005</anyo>

</edicion>

</libro>

<libro>

<autor>Cody Lindley.</autor>

<titulo> jQuery </titulo>

<numedicion>4</numedicion>

<edicion numedic="4">

<anyo>2004</anyo>

</edicion>

</libro>

</datos>

</operaciones>

</catalogo>

**XSLT**

<?xml version="1.0" encoding="ISO-8859-1"?>

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

<xsl:variable name="hoy">

2005

</xsl:variable>

<xsl:template match="//catalogo">

<html lang="es-ES">

<head>

<meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>

<title>

Inteligencia de Negocios

</title>

</head>

<body style="font-family : sans-serif;">

<h1>

Inteligencia de negocios - "Clasificador"

</h1>

<h2>

Mejores libros informaticos de

<xsl:value-of select="$hoy - 5">

</xsl:value-of>

o posteriores, ordenadas por autor

</h2>

<table style="border : 2px solid black; border-collapse : collapse" summary="Listado tabulado de las obras">

<tr style="color : black; background-color : green;">

<th style="border : 1px solid green;">

Autor

</th>

<th style="border : 1px solid green;">

Título

</th>

</tr>

<xsl:for-each select="operaciones/datos/libro[edicion/anyo >= $hoy - 5]">

<xsl:sort select="autor[1]">

</xsl:sort>

<tr>

<td style="border : 1px solid green;">

<xsl:value-of select="autor">

</xsl:value-of>

</td>

<td style="border : 1px solid green;">

<xsl:apply-templates select="titulo" />

<xsl:if test="edicion[@numedic]">

(

<xsl:value-of select="edicion/@numedic" />

a. ed.)

</xsl:if>

<xsl:choose>

<xsl:when test="edicion/anyo >= $hoy - 1"> (\*\*\*\*) </xsl:when>

<xsl:when test="edicion/anyo >= $hoy - 2"> (\*\*\*) </xsl:when>

<xsl:when test="edicion/anyo >= $hoy - 3"> (\*\*) </xsl:when>

<xsl:when test="edicion/anyo >= $hoy - 4"> (\*) </xsl:when>

<xsl:otherwise>

</xsl:otherwise>

</xsl:choose>

</td>

<td>

<td style="border : 1px solid gray;">

<xsl:apply-templates select="anyo">

</xsl:apply-templates>

<xsl:if test="edicion/anyo = 2005">

(2005)

</xsl:if>

</td>

</td>

</tr>

</xsl:for-each>

</table>

<p>

Nota: Los asteriscos indican obras de los cuatro últimos años (\*\*\*\*: año en curso -

<xsl:value-of select="$hoy"/>

- y anterior)

</p>

<h1>

Lista de consultas SQL

</h1>

<xsl:for-each select="child::\*/\*">

<xsl:choose>

<xsl:when test="name()='disenio'">

<xsl:for-each select="child::\*">

CREATE TABLE

<xsl:value-of select="name(.)"/>

(

<xsl:for-each select="child::\*">

<xsl:value-of select="name(.)"/>

<xsl:choose>

<xsl:when test="(.)='entero'"> int </xsl:when>

<xsl:when test="(.)='texto'"> text </xsl:when>

<xsl:when test="(.)='logico'"> boolean </xsl:when>

<xsl:when test="(.)='decimal'"> double </xsl:when>

<xsl:when test="(.)='fecha'"> date </xsl:when>

</xsl:choose>

<xsl:if test="following-sibling::\*">

,

</xsl:if>

</xsl:for-each>

);

<br/>

</xsl:for-each>

</xsl:when>

<xsl:when test="name()='datos'">

<xsl:for-each select="child::\*">

INSERT INTO

<xsl:value-of select="name(.)" />

VALUES (

<xsl:for-each select="child::\*">

<xsl:value-of select="(.)" />

<xsl:if test="following-sibling::\*">

,

</xsl:if>

</xsl:for-each>

);

<br/>

</xsl:for-each>

</xsl:when>

</xsl:choose>

</xsl:for-each>

</body>

</html>

</xsl:template>

<xsl:template match="enfasis">

<span style="font-weight : bold;">

<xsl:value-of select=".">

</xsl:value-of>

</span>

</xsl:template>

</xsl:stylesheet>

**VISTA FINAL**



**Repositorio en GitHub:** <https://github.com/fatalityignpab/XML-XSLT-ETS>