**Natalia Maya Toro**

**Taller Sobre XML - XSL**

**Profesor Hernando Recaman Chaux**

|  |
| --- |
|  |

**Politécnico Colombiano Jaime Isaza Cadavid**

**Asignatura de Programación Distribuida y Paralela**

**Medellín**

**PROYECTO PARA DOS ESTUDIANTES**

**Fecha de Entrega: septiembre 27 de 2022**

1. Se tiene información sobre el presupuesto del Politécnico Colombiano JIC en millones de pesos así: (40 puntos)

**Concepto 2019 2020 2021 2022 2023**

Ingresos operacionales

Matriculas 900 300 110 120 180

Derechos pecuniarios 890 130 200 380 130

EGRESOS OPERACIONALES

Servicios 690 140 108 130 120

Gastos 790 110 112 180 105

……

Utilizar XML y XSL para estructurar y crear un listado o tabla con las siguientes características:

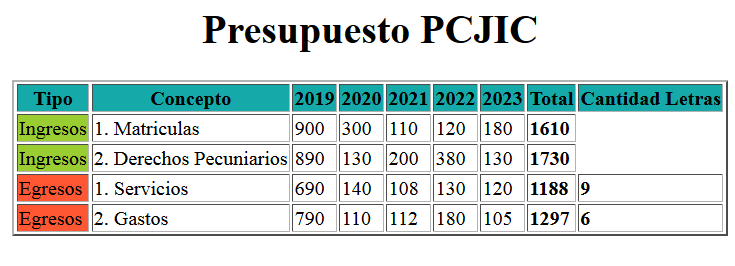
1. Informar la sumatoria total de cada rubropara los ingresos y egresos operacionales (matriculas, Derechos pecuniarios, Servicios personales …)

2. Informar la cantidad de letras que tiene cada uno de los conceptos de tipo egresos, (Servicios-9, Gastos-6).

3. Informa el total de Ingresos y egresos operacionales para todos los años.

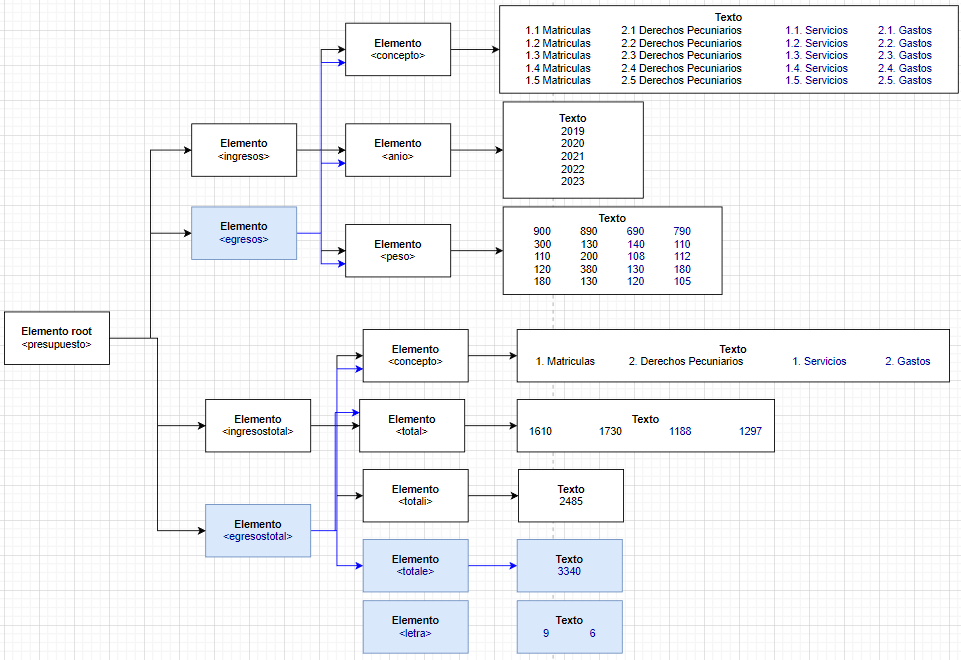
4. Enumerar cada tipo de concepto por aparte (ingresos y egresos).

5. Modelo del Árbol XML.

****

**Totales 3270 680 …**

3. Aprendizaje independiente (10 puntos)

**Modelo del árbol XML**

**XML**

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

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

<presupuesto>

<ingresos>

<concepto>1.1 Matriculas</concepto>

<anio>2019</anio>

<peso>900</peso>

</ingresos>

<ingresos>

<concepto>1.2 Matriculas</concepto>

<anio>2020</anio>

<peso>300</peso>

</ingresos>

<ingresos>

<concepto>1.3 Matriculas</concepto>

<anio>2021</anio>

<peso>110</peso>

</ingresos>

<ingresos>

<concepto>1.4 Matriculas</concepto>

<anio>2022</anio>

<peso>120</peso>

</ingresos>

<ingresos>

<concepto>1.5 Matriculas</concepto>

<anio>2023</anio>

<peso>180</peso>

</ingresos>

<ingresostotal>

<concepto>1 Matriculas</concepto>

<total>1610</total>

</ingresostotal>

<ingresos>

<concepto>2.1 Derechos Pecuniarios</concepto>

<anio>2019</anio>

<peso>890</peso>

</ingresos>

<ingresos>

<concepto>2.2 Derechos Pecuniarios</concepto>

<anio>2020</anio>

<peso>130</peso>

</ingresos>

<ingresos>

<concepto>2.3 Derechos Pecuniarios</concepto>

<anio>2021</anio>

<peso>200</peso>

</ingresos>

<ingresos>

<concepto>2.4 Derechos Pecuniarios</concepto>

<anio>2022</anio>

<peso>380</peso>

</ingresos>

<ingresos>

<concepto>2.5 Derechos Pecuniarios</concepto>

<anio>2023</anio>

<peso>130</peso>

</ingresos>

<ingresostotal>

<concepto>2. Derechos Pecuniarios</concepto>

<total>1730</total>

</ingresostotal>

<ingresostotal>

<totali>2485</totali>

</ingresostotal>

<egresos>

<concepto>1.1 Servicios</concepto>

<anio>2019</anio>

<peso>690</peso>

</egresos>

<egresos>

<concepto>1.2 Servicios</concepto>

<anio>2020</anio>

<peso>140</peso>

</egresos>

<egresos>

<concepto>1.3 Servicios</concepto>

<anio>2021</anio>

<peso>108</peso>

</egresos>

<egresos>

<concepto>1.4 Servicios</concepto>

<anio>2022</anio>

<peso>130</peso>

</egresos>

<egresos>

<concepto>1.5 Servicios</concepto>

<anio>2023</anio>

<peso>120</peso>

</egresos>

<egresostotal>

<concepto>1. Servicios</concepto>

<total>1188</total>

<letra>9</letra>

</egresostotal>

<egresos>

<concepto>2.1 Gastos</concepto>

<anio>2019</anio>

<peso>790</peso>

</egresos>

<egresos>

<concepto>2.2 Gastos</concepto>

<anio>2020</anio>

<peso>110</peso>

</egresos>

<egresos>

<concepto>2.3 Gastos</concepto>

<anio>2021</anio>

<peso>112</peso>

</egresos>

<egresos>

<concepto>2.4 Gastos</concepto>

<anio>2022</anio>

<peso>180</peso>

</egresos>

<egresos>

<concepto>2.5 Gastos</concepto>

<anio>2023</anio>

<peso>105</peso>

</egresos>

<egresostotal>

<concepto>2. Gastos</concepto>

<total>1297</total>

<letra>6</letra>

</egresostotal>

<egresostotal>

<totale>3340</totale>

</egresostotal>

</presupuesto>

**XSL**

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

<xsl:stylesheet version="1.1" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/" >

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>PRESUPUESTO POLI</title>

</head>

<body>

<table border="1" style="float: left;" width="50%">

<td><h1>INGRESOS</h1></td>

<tr bgcolor="#9acd32">

<th><h3>CONCEPTO</h3></th>

<th><h3>AÑO</h3></th>

<th><h3>PESOS</h3></th>

</tr>

<xsl:for-each select="//presupuesto/ingresos">

<tr>

<th><xsl:value-of select="concepto"/></th>

<th><xsl:value-of select="anio"/></th>

<th><xsl:value-of select="peso"/></th>

</tr>

</xsl:for-each>

</table>

<table border="1" width="50%">

<td><h1>EGRESOS</h1></td>

<tr bgcolor="#3396FF">

<th><h3>CONCEPTO</h3></th>

<th><h3>AÑO</h3></th>

<th><h3>PESOS</h3></th>

</tr>

<xsl:for-each select="//presupuesto/egresos">

<tr>

<th><xsl:value-of select="concepto"/></th>

<th><xsl:value-of select="anio"/></th>

<th><xsl:value-of select="peso"/></th>

</tr>

</xsl:for-each>

</table>

<table border="1" style="float: left;" width="50%">

<td><h1>TOTAL DE INGRESOS</h1></td>

<tr bgcolor="#B5FF33">

<th><h3>CONCEPTO</h3></th>

<th><h3>TOTAL POR CONCEPTO</h3></th>

<th><h3>TOTAL</h3></th>

</tr>

<xsl:for-each select="//presupuesto/ingresostotal">

<tr>

<th><xsl:value-of select="concepto"/></th>

<th><xsl:value-of select="total"/></th>

<th><xsl:value-of select="totali"/></th>

</tr>

</xsl:for-each>

</table>

<table border="1" width="50%">

<td><h1>TOTAL EGRESOS</h1></td>

<tr bgcolor="#33FFF3">

<th><h3>CONCEPTO</h3></th>

<th><h3>CANTIDAD DE LETRAS</h3></th>

<th><h3>TOTAL POR CONCEPTO</h3></th>

<th><h3>TOTAL</h3></th>

</tr>

<xsl:for-each select="//presupuesto/egresostotal">

<tr>

<th><xsl:value-of select="concepto"/></th>

<th><xsl:value-of select="letra"/></th>

<th><xsl:value-of select="total"/></th>

<th><xsl:value-of select="totale"/></th>

</tr>

</xsl:for-each>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

**Aprendizaje independiente (10 puntos)**

**XML**

**¿De dónde viene?**

XML (eXtensible Markup Language) comenzó a desarrollarse en septiembre de 1996 auspiciado por el W3C para diseñar un lenguaje de marcado optimizado y hoy es el estándar internacional para intercambio de información, multiplataforma y multi-entorno, combinando la simplicidad de HTML.

En pocos años, XML se ha convertido en el lenguaje con mayor impacto en el desarrollo de aplicaciones, promovido por Microsoft, IBM, Sun, entre otras empresas de software.

**¿Qué es?**

XML es un Meta-Lenguaje abierto, que permite la definición de lenguajes de representación de documentos, para su uso en la WWW, permite describir el sentido o la semántica de los datos a diferencia del HTML, separa el contenido de la presentación. XML permite crear una estructura de marca con absoluta libertad para estructurar información, la flexibilidad y la potencia de esa estructura permite definir información con una riqueza enorme (Objetos completos, imágenes, procesos complejos, entre otros.) Esa información se expresa en texto, entendible para cualquier procesador.

**Características**

• La posibilidad de marcado descriptivo, con un conjunto de marcas abierto. En XML las marcas diferencian los contenidos informativos de los documentos, frente al uso que se hace en HTML para visualizar contenidos.

• XML no especifica un conjunto válido de marcas, sino que nos ofrece reglas que nos permiten crear nuevos conjuntos de marcas.

• Hacer explícita su estructura y sus contenidos informativos.

• Crear documentos que puedan intercambiarse y procesarse con facilidad en aplicaciones heterogéneas.

• Cada documento XML posee una estructura lógica y una física.

• La estructura lógica es una serie de declaraciones, elementos, comentarios, entre otros, que se indican en el documento mediante marcas explícitas.

• La estructura física del documento es una serie de unidades llamadas entidades, que indica los datos que contendrá el documento. Las estructuras lógica y física deben anidarse de forma correcta.

**XSL**

XSL (eXtensible Stylesheet Language), según el W3C, XSL es "un lenguaje para transformar documentos XML", así como un vocabulario XML para especificar semántica de formateo de documentos. En definitiva, además del aspecto que ya incluía la presentación y estilo de los elementos del documento, añade una pequeña sintaxis de lenguaje de script para poder procesar los documentos XML de forma más cómoda.

Al igual que con HTML hasta ahora, se pueden especificar las hojas de estilo, sean CSS o XSL, dentro del propio documento XML o referenciándolas de forma externa. XSL se estructura en base al concepto de plantilla de trabajo, que básicamente es un conjunto de instrucciones modelo, entremezcladas con etiquetas XSL que repiten bloques de instrucciones para cada nodo seleccionado, de un documento XML. Existen etiquetas adicionales que sustituyen el valor de los nodos especificados en las instrucciones modelo; lo que se trata de hacer es acoplar los diferentes datos que existen en un documento XML, transformado en un esquema que reconozca tanto la sintaxis de HTML como XML.

**XSLT**

XSLT (Extensible Stylesheet Language Transformation), permite que podamos darle un estilo de presentación a los documentos XML, es decir que los datos también pueden tener una capa de presentación, ajena a su funcionalidad como data misma. Un archivo XSLT nos permite transformar un documento XML de un formato de presentación a otro. El modelo de procesamiento XSLT consiste en la técnica siguiente:

**1.** El proceso inicia con el recorrido del árbol del documento, buscando una coincidencia con una regla para cada nodo que se visita.

**2.** Una vez que se encuentra una regla el cuerpo de esta se convierte en un documento. Las instrucciones de procesamiento XSLT dentro de la regla usan al nodo que coincide a su contexto.

**3.** Si se desea procesamiento adicional, se debe incluir la instrucción xsl:apply– templates. Los nodos que habrán de procesarse se encuentran especificados mediante el atributo match. Si se omite éste, el comportamiento predeterminado consiste en procesar los elementos secundarios del nodo actual.

**Webgrafía**

*ANÁLISIS COMPARATIVO DE TECNOLOGÍAS PARA DESARROLLAR APLICACIONES DINÁMICAS DE INTERNET ENRIQUECIDAS (RIA --RICH INTERNET APPLICATION)*. (n.d.). Edu.Ec. Retrieved September 27, 2022, from http://repositorio.utn.edu.ec/bitstream/123456789/581/3/CAPIII.pdf

Antonio, J., & Burgos, E. (n.d.). *XML (extensible markup language)*. Rclis.org. Retrieved September 27, 2022, from http://eprints.rclis.org/4124/1/apuntes\_xml.PDF

Rodríguez Muñoz, J. V., Manuel, P., & Ortuño, D. (n.d.). *Arquitectura de la información: XML y WEB*. Core.ac.uk. Retrieved September 27, 2022, from https://core.ac.uk/download/pdf/38981317.pdf