



## Practica No. 6

### Introducción a XML

Nombre(s):

Martin Isai Nuñez Villeda

Paola Monserrat Osorio

#### Objetivo:

Con la realización de esta práctica se pretende: implementar en una página web que emplee un archivo XML.

#### Fundamento Teórico:

##### XML

XML son las siglas de **eXtensible Markup Language**, es un formato diseñado para almacenar y transportar datos independiente de cualquier software o hardware.

Un archivo XML se divide en dos partes: *prolog* y *body*.

La parte **prolog** consiste en metadatos administrativos, como declaración XML, instrucción de procesamiento opcional, declaración de tipo de documento y comentarios.

La parte del **body** se compone de dos partes: estructural y de contenido (presente en los textos simples).

El diseño XML se centra en la simplicidad, la generalidad y la facilidad de uso y, por lo tanto, se utiliza para varios servicios web.

Tanto es así que hay sistemas destinados a ayudar en la definición de lenguajes basados en XML, así como APIs que ayudan en el procesamiento de datos XML - que no deben confundirse con HTML.

Es parecido a HTML y fue diseñado para ser auto-descriptivo, es decir, que por sí mismo indique las características principales y las posibles operaciones que se puedan realizar.

En qué XML no se parece a HTML

- XML está diseñado para almacenar datos, enfocado a dato
- HTML está diseñado para desplegar datos, enfocado en la apariencia de los datos

#### Forma de trabajo:

Colaborativa en equipos de 3 personas

#### Material:

1. Computadora
2. IDE Netbeans

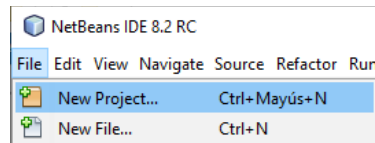
### Procedimiento:

El objetivo es crear una aplicación web que interactue con un archivo en formato XML.

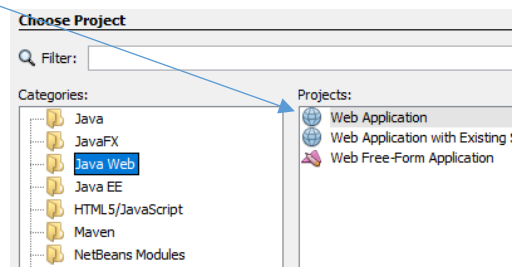
#### Desarrollo

Para la creación de la página web deberán realizarse los siguientes pasos:

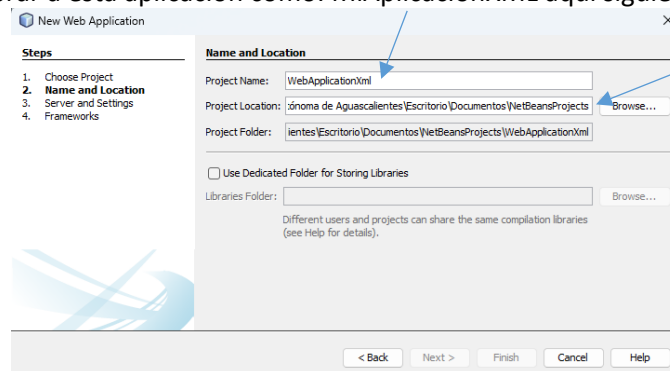
1. Abrir el IDE Netbeans.
2. crear un proyecto nuevo



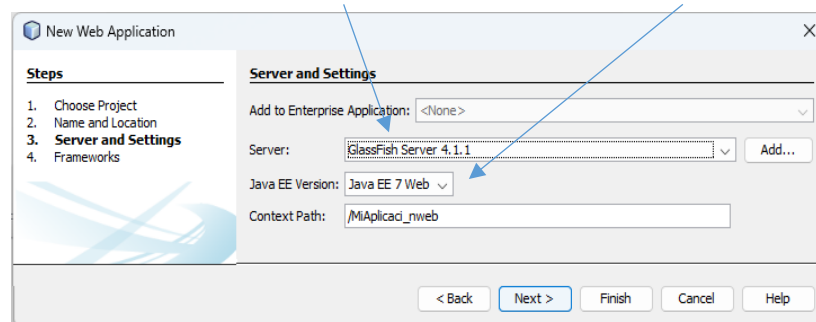
3. Seleccionar Java Web > Web Application



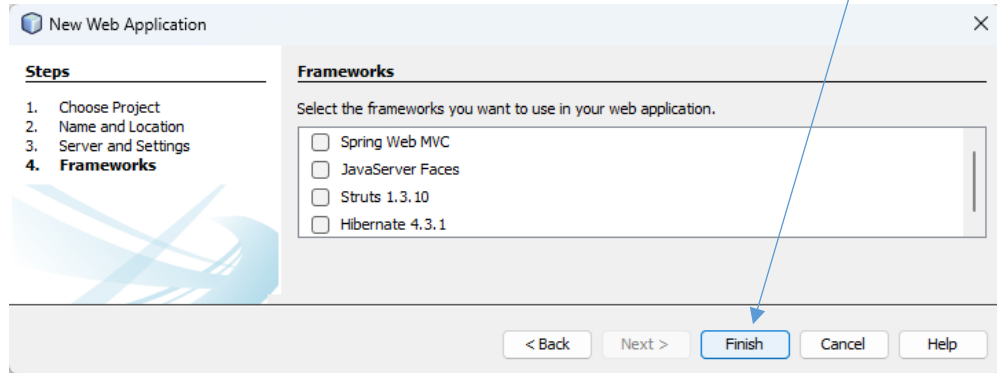
4. Se va a nombrar a esta aplicación como: MiAplicacionXML aquí siguiendo este camino



5. Se debe seleccionar al servidor GlassFish y la versión Java EE 7 web.



6. Para esta aplicación no se va a seleccionar una estructura y se termina:



7. A continuación, crea con el block de notas un archivo XML con el siguiente contenido de ejemplo (lo azul) y nómbralo como cd\_catalog.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<CATALOG>
<CD>
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
</CD>
<CD>
<TITLE>Hide your heart</TITLE>
<ARTIST>Bonnie Tyler</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>CBS Records</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1988</YEAR>
</CD>
</CATALOG>
```

La primera línea solo es el identificador de archivos XML no almacena nada.

<CATALOG>	es el identificador primario del archivo
<CD>	es un renglón
<TITLE> <ARTIST> <COUNTRY> ...<YEAR> etc	son columnas

**8. Ahora agrega el código javascript para leer archivos**

```
<!DOCTYPE html>
<html>
  <head>
    <style>
table {border-collapse:separate;
  width: 70%;
  box-shadow: 5px 5px 10px 10px;}
th, td {text-align: left;
  padding: 12px;}
tr:nth-child(even){background-color: #f2f2f2}
    </style>
  </head>
  <body>
<p><button onclick="loadXMLDoc()"> Get CD info </button></p>
<table id="demo" >
<tr><th>Artist</th><th>Title</th></tr>
</table>
<script>
function loadXMLDoc() {
  var xmlhttp = new XMLHttpRequest();
  xmlhttp.onreadystatechange = function()
  {if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
    {myFunction(xmlhttp)};};
  xmlhttp.open("GET", "cd_catalog.xml" , true);
  xmlhttp.send();}
function myFunction(xml)
{ var x, i, xmlDoc, table;
  xmlDoc = xml.responseXML;
  table = "<tr> <th>Artist</th> <th>Title</th> </tr>";
  x = xmlDoc.getElementsByTagName("CD");  // tag del xml  ARTIST TITLE
  for (i = 0; i < x.length; i++)
  { table += "<tr><td>" +
    x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue + "</td><td>" +
    x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue + "</td></tr>";}
  document.getElementById("demo").innerHTML = table;}
</script>
</body>
</html>
```

Al principio del archivo index se crea en el archivo HTML con un botón que genere un evento para que lea el archivo, algo así

```
<button onclick="loadXMLDoc()"> Get CD info </button>
```

Luego se crea la **table** donde va a aparecer la información, y se le asigna un ID=demo, así

```
<table id="demo" >
```

```
<tr><th>Artist</th><th>Title</th></tr>
```

```
</table>
```

La función script, que van a ser 2 funciones

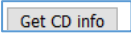
1. La función **loadXMLDOC** que almacenará en un objeto la información de una petición HTTP XML, la variable donde se almacena la información es xmlhttp
2. la función **MyFunction()** que será la encargada de generar código HTML dinámicamente para hacer una table y desplegar el contenido de la variable xmlhttp

En esta función la instrucción para obtener una columna es

```
x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue
```

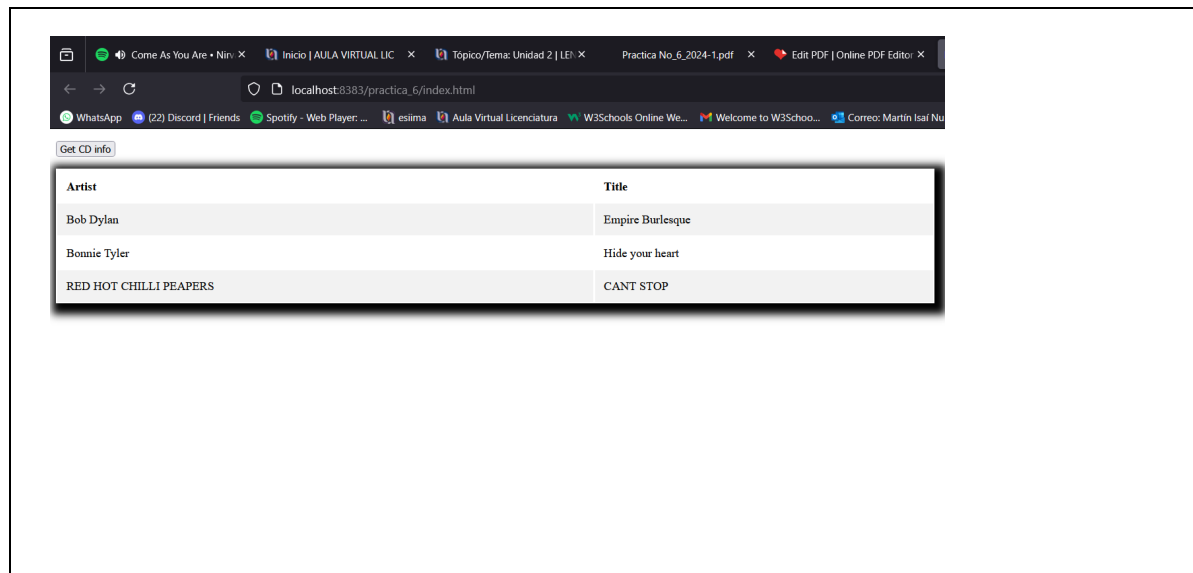
donde **ARTIST** debe existir en el archivo XML, toda la información se guarda en una variable string llamada **table**, junto con instrucciones HTML dinámico, al final la variable **table** se inserta en la tabla HTML llamada "demo" así

```
document.getElementById("demo").innerHTML = table;
```

El código que incluye el <style> igual al ejemplo de HTML dinámico, para desplegar información dando un click en el botón. 

## Resultados:

Realizar una corrida de prueba y mostrar cómo se ve la página en el navegador.





Una vez terminados los programas (.html) y (.xml) debe subirse a la plataforma de **aulavirtual** junto con este reporte.

### ***Conclusiones:***

En esta práctica de HTML, aprendimos a crear y leer archivos XML, permitiendo estructurar datos de manera organizada.

Implementamos su lectura y visualización en una página web mediante JavaScript y DOM, mejorando la presentación de la información.

Esto nos permitió comprender la importancia de XML para el intercambio de datos en la web. En conclusión, esta práctica refuerza el manejo de datos estructurados y su integración con tecnologías web.