GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura i Esport

ceedcv CENTRE ESPECÍFIC D'EDUCACIÓ A DISTÀNCIA DE LA COMUNITAT VALENCIANA

Tema 1. ACCESO A EXPEDIENTES Parte 2. Expedientes. XML y XSL

Acceso a Datos (ADA) (a distancia en inglés)

CFGS Desarrollo de Aplicaciones Multiplataforma (DAM)

Abelardo Martínez

Año 2023-2024

Créditos

•Apuntes realizados por Abelardo Martínez. •Basado y modificado de Sergio Badal (www.sergiobadal.com). •Las imágenes e iconos utilizados están protegidos por la LGPL . licencia y haber sido de:

- •https://commons.wikimedia.org/wiki/Crystal_Clear
- •https://www.openclipart.org

Progreso de la unidad

Contenido

1. ¿QUÉ ES XML?

lenguaje XML

XML es una herramienta/lenguaje independiente de software y hardware para almacenar y transportar datos.

Principales características:

1) XML significa lenguaje de marcado extensible
2) XML es un lenguaje de marcado muy parecido a HTML
3) XML fue diseñado para almacenar y transportar datos (como lo hace JSON o CSV)
4) XML fue diseñado para ser autodescriptivo (intuitivo)
5) XML es una recomendación del W3C (estándar)
6) XML es una forma restringida de SGML, el lenguaje de marcado generalizado estándar [ISO 8879]

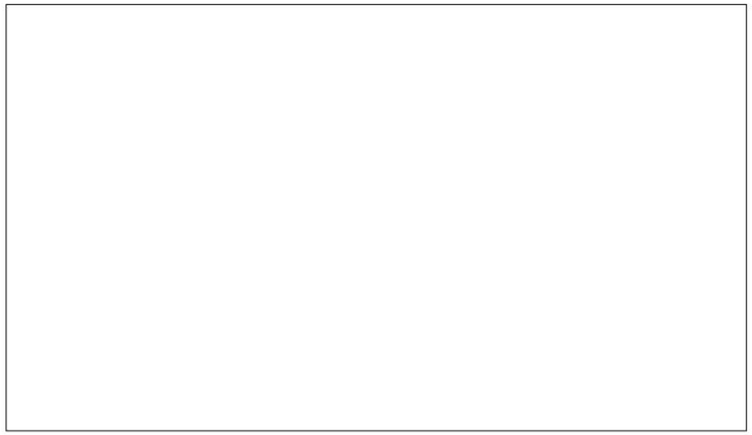
Más información sobre XML:

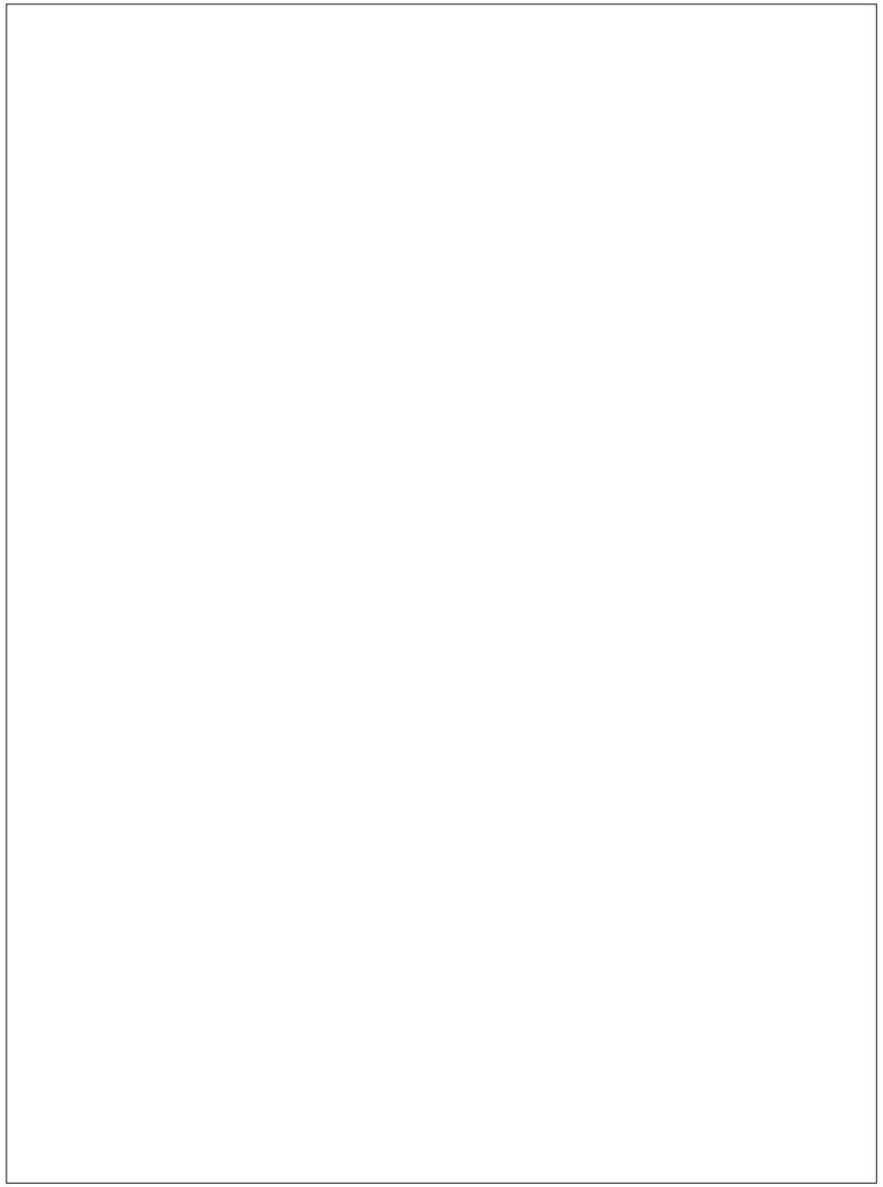
https://www.w3schools.com/xml/

La diferencia entre XML y HTML

XML y HTML fueron diseñados con diferentes objetivos:

- XML fue diseñado para transportar datos , centrándose en lo que los datos son.
- HTML se diseñó para mostrar datos, centrándose en cómo miradas de datos.
- Las etiquetas XML no están predefinidas como las etiquetas HTML.

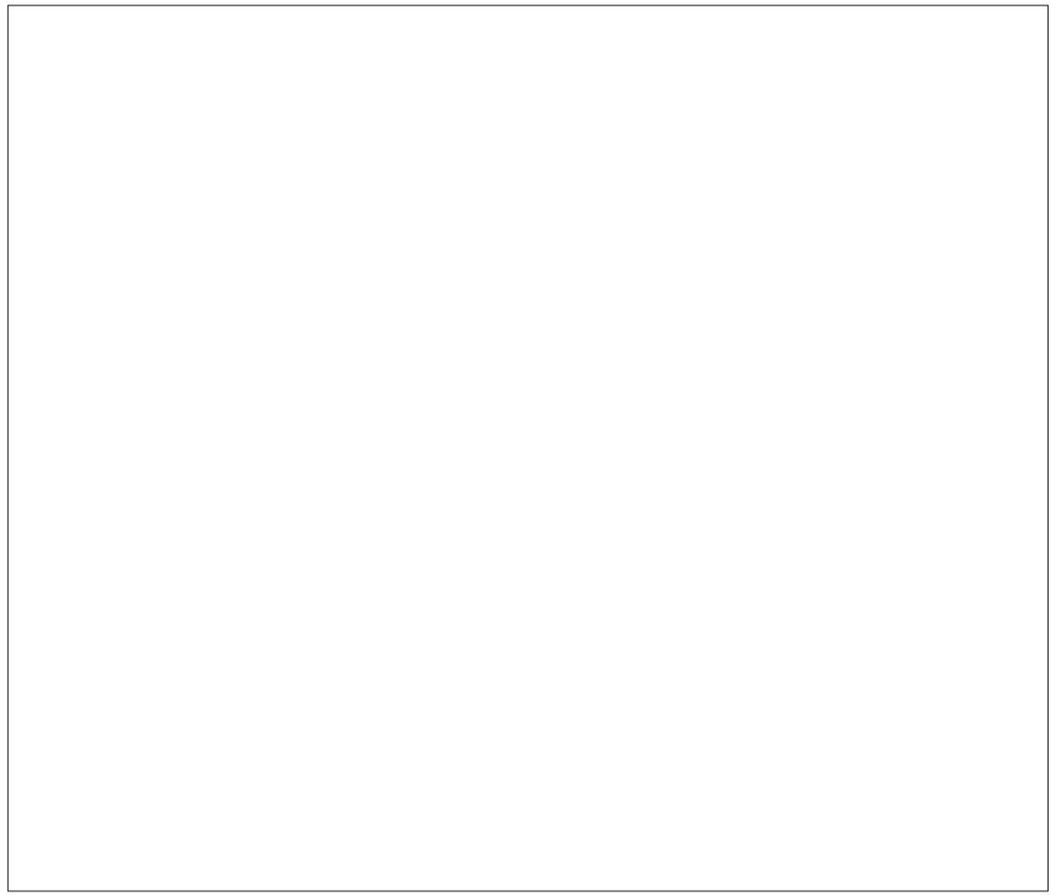




La diferencia entre XML y JSON

XML y JSON son dos formas similares de almacenar información.

Más información sobre JSON:
https://www.w3schools.com/whatis/whatis_json.asp



2. TRABAJAR CON ARCHIVOS XML

Trabajar con archivos XML

Los archivos XML se pueden utilizar para:

• Proporcionar datos a una base de datos.

• Almacenar información en bases de datos

especiales. • Archivos

de configuración • Intercambio de información en entornos web

(SOAP). • Ejecución de comandos en servidores remotos.

• Etc.

Para realizar cualquiera de estas operaciones es necesario un lenguaje de programación que proporcione funcionalidad a XML (XML no es un lenguaje de Turing completo).

Para acceder a archivos XML, leer su contenido o modificar su estructura, se utiliza un procesador XML o un analizador XML .

Hay dos formas principales de trabajar con archivos XML utilizando Java: • SAX

• DOM

analizadores XML

1) SAX (API simple para análisis XML)

Un analizador SAX es una rutina basada en eventos para analizar el XML mediante devoluciones de llamada. SAX puede aislar datos XML en una única lectura secuencial detectando los hashtags de apertura y cierre.

Ventajas:

- Es muy rápido.
- NO carga todo el documento en memoria.
- La mejor opción para leer documentos grandes.

Desventajas:

- Tiene que leer el documento completo en cada consulta.

2) DOM (modelo de objetos de documento)

DOM almacena todos los datos del documento XML en memoria en una estructura de árbol jerárquica.

Ventajas:

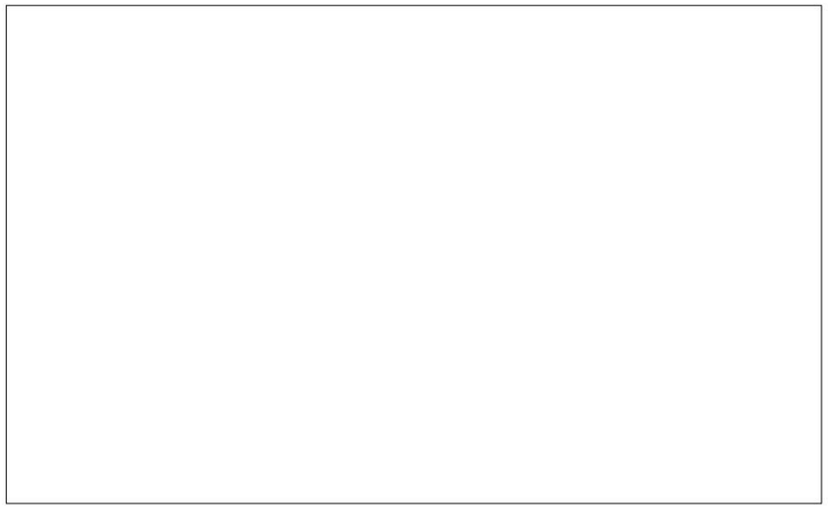
- En pocas palabras, un analizador DOM funciona en todo el documento XML.
- Es ideal para aplicaciones que requieren consulta continua de datos.

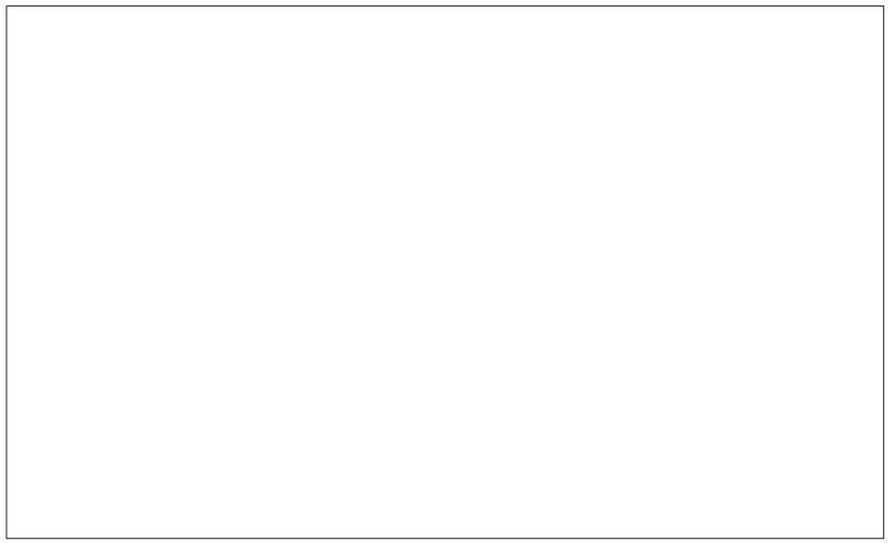
Desventajas:

- Carga todo el documento en la memoria.
- No apto para documentos grandes.

SAX versus DOM

Más información sobre XML y Java:
https://www.baeldung.com/java-xml





3. SAXO. LEER ARCHIVOS XML CON JAVA

Acceso a archivos con SAX

La API SAX está completamente incluida en el JRE.

• Si le damos el archivo XML a la API, no hará nada. • Al cargar, se llamarán algunos métodos para cada elemento encontrado.

• Todos estos métodos están definidos en DefaultHandler.

Créditos y código (Cómo leer un archivo XML en Java (SAX Parser): https://mkyong.com/java/how-to-read xml-file-in-java-sax-parser/

Acceso a archivos con SAX

Pasos para leer un archivo XML con SAX:

- Lo primero que haremos será importar todas las clases e interfaces necesarias.
- Crearemos una clase para extender DefaultHandler.
- Luego, anularemos startDocument, endDocument, startElement, endElement y Métodos de personajes .
- Con esto podremos imprimir todos los elementos XML, atributos, comentarios y textos.

Versión de Java:

Acceso a archivos con SAX

Producción:

- Este podría ser el resultado de colocar un println simple en cada método, pero podríamos almacenar la información en una base de datos y guardarla en otra.
  dispositivo o enviarlo a otro lugar.

4. DOMINGO. GESTIONAR ARCHIVOS XML CON JAVA

Acceso a archivos con DOM

• El analizador DOM está diseñado para trabajar con XML como objeto. • Este objeto es un gráfico (una estructura similar a un árbol) en la memoria, llamado “Modelo de objetos de documento (DOM)”.

• El procedimiento es bastante similar al utilizado en JavaScript:

1) Primero, el analizador recorre todo el archivo XML creando objetos DOM correspondientes a los nodos del archivo XML.

Estos objetos DOM están vinculados entre sí en un árbol como estructura.

2) Una vez que el analizador termina con el proceso de análisis, obtenemos esta estructura de objetos DOM en forma de árbol.

Ahora podemos “recorrer” la estructura DOM de un lado a otro como queremos obtener/actualizar/eliminar datos del mismo.

Créditos y código (ejemplo de analizador DOM de Java): https://howtodoinjava.com/java/xml/read-xml-dom-parser-example/

4.1 Leer archivos XML con DOM

Lectura de archivos con DOM

Pasos para leer un archivo XML con DOM: •

Crear DocumentFactory (cargar XML en la memoria) • Crear DocumentBuilder

• Crear objeto de documento •

Extraer elemento raíz

•Recorrer el documento Examinar los atributos

Examinar subelementos

Repetir recursivamente

Créditos y código (ejemplo de analizador DOM de Java): https://howtodoinjava.com/java/xml/read-xml-dom parser-example/

Lectura de archivos con DOM

Ejemplo completo:

Versión de Java:

Lectura de archivos con DOM

Producción:

- Este podría ser el resultado de colocar un println simple en cada elemento, pero podríamos almacenar la información en una base de datos y guardarla en otra.

dispositivo o enviarlo a otro lugar.

4.2 Escribir archivos XML con DOM

Escritura de archivos con DOM

Pasos para crear y escribir XML en un archivo: •

Crear un documento (cargar XML en la memoria). • Crear

elementos XML, atributos, etc., y

adjuntar al documento doc.

• Crear un transformador para escribir el documento.

al disco.

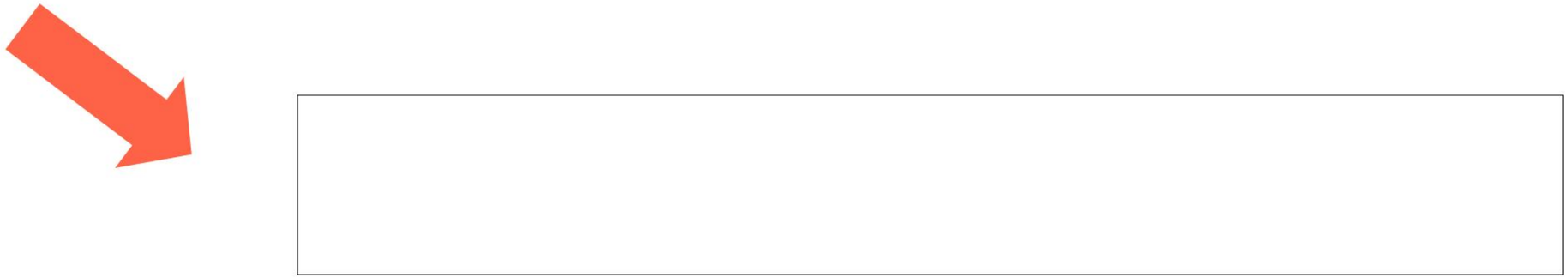
Créditos y código (Crear archivo XML en java DOM):

https://mkyong.com/java/how-to-create-xml-file-in-j ava-dom/

1) Creación (cargar XML en la memoria):

Escritura de archivos con DOM

2) Escritura:



4.3 Modificar estructura XML con DOM

Modificación de archivos con DOM

Pasos para cambiar varios valores sólo dentro de nodos específicos: •

Crear un documento

(cargar XML en la memoria). • Bucle para todos los subnodos. • Si

se encuentran nodos específicos,

cambie los valores de

sus subnodos.

• Guardar XML en el disco.

Créditos y código (Actualizar nodos y atributos con DOM): https://

learningprogramming.net/java/dom/update-n odes-and-attributes-with-dom-in-java-xml/

1) Creación (cargar XML en la memoria):

Modificación de archivos con DOM

2) Tutorial de la estructura XML:

3) Actualizar cambios y guardar:

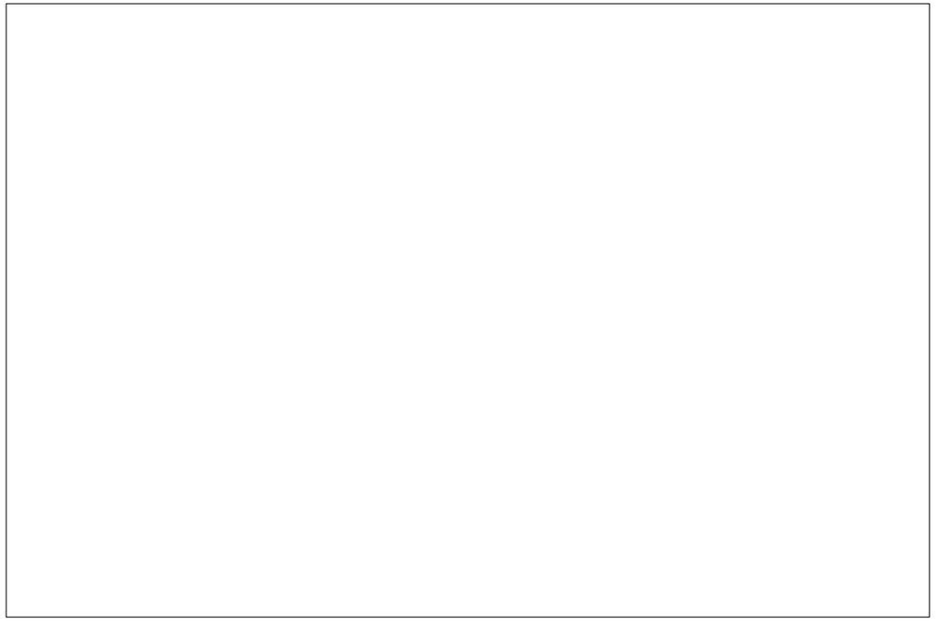
5. ¿QUÉ ES XSL?

lenguaje XSL

XSL es un lenguaje de estilo para
XML: • XSL significa lenguaje de hoja de estilo
extensible . • XSL es un lenguaje de estilo para XML al igual que CSS es un lenguaje
de estilo para HTML. • XSLT significa Transformación XSL y se utiliza para transformar documentos XML a otros formatos (XML => HTML)

Más información sobre XSL:

https://www.w3schools.com/xml/xsl_intro.asp
https://www.tutorialspoint.com/xslt/index.htm



¿Cuál es el nivel requerido de XSL?

XSL/XSLT puede ser tan complejo como CSS.

Aprender XSL en profundidad NO es el objetivo de esta UNIDAD.

Nuestro propósito es:

• Comprender qué hace XSL/XSLT. • Aprenda
algunos conceptos básicos de XSL/XSLT.

• Aprenda cómo aplicar un archivo XSL a un archivo XML para obtener un HTML usando Java.

Presentaremos los conceptos básicos de XSL/Java trabajando con este ejemplo: http://
javaonlineguide.net/2016/02/convert-xml-to-html-in-java-using-xslt-example.html Puedes ir más allá en XSL/Java
estudiando este ejemplo: https://www.oreilly.com/library/view/learning-
java-4th/9781449372477/ch24s10.html

Trabajar con XSL

Presentemos un ejemplo para comprender qué hace realmente XSLT.
Créditos y código (ejemplo básico
XSLT): https://developer.mozilla.org/en-US/docs/Web/API/XSLTProcessor/Basic_Example

ejemplo.xml

producción

ejemplo.xsl

6. USAR XSL PARA TRANSFORMAR DOCUMENTOS

Convierta XML a HTML en Java usando XSLT

En este ejemplo, convertiremos XML a HTML usando el lenguaje XSLT.

1) Primero, echemos un vistazo al XML. Como puedes ver, es muy simple y llanamente.

Créditos y código (Convertir XML a HTML en Java usando XSLT): http://javaonlineguide.net/2016/02/convert-xml-to-html-in-java-using-xslt-example.html

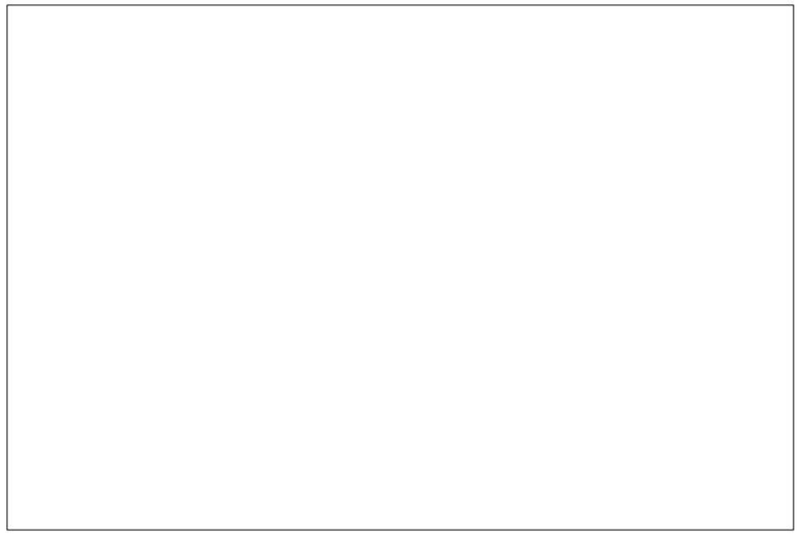
XML

Convierta XML a HTML en Java usando XSLT

En este ejemplo, convertiremos XML a HTML usando el lenguaje XSLT.

2) En segundo lugar, estudiemos qué tipo de resultado (HTML) queremos obtener.

XML



¿XSL?

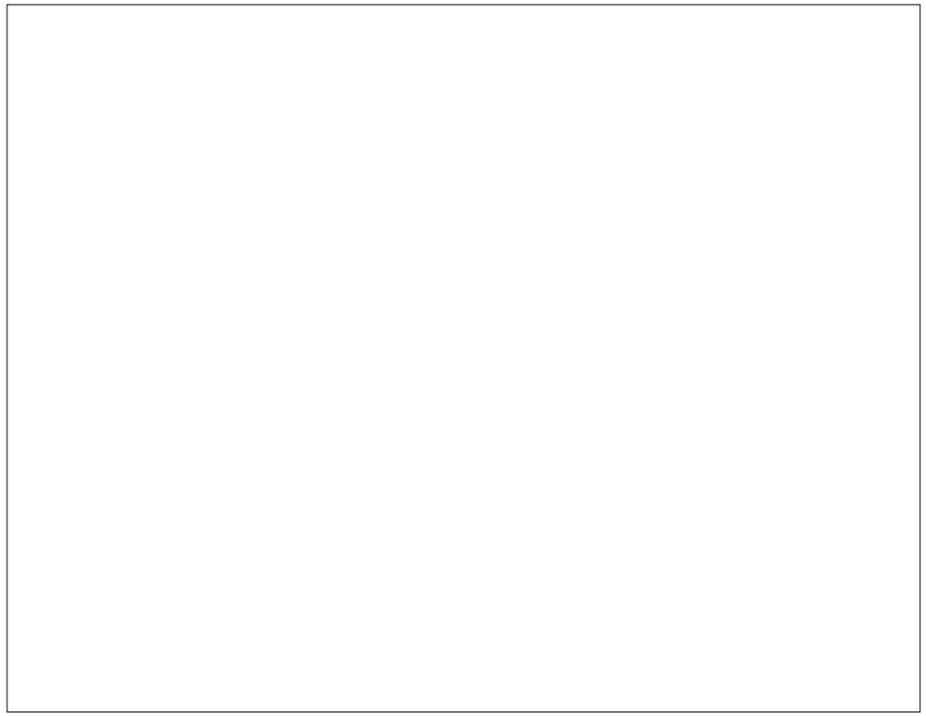
Salida (HTML)

Convierta XML a HTML en Java usando XSLT

En este ejemplo, convertiremos XML a HTML usando el lenguaje XSLT.

2) En segundo lugar, estudiemos qué tipo de resultado (HTML) queremos obtener.

HTML



Salida (HTML)

¿Y el CSS?

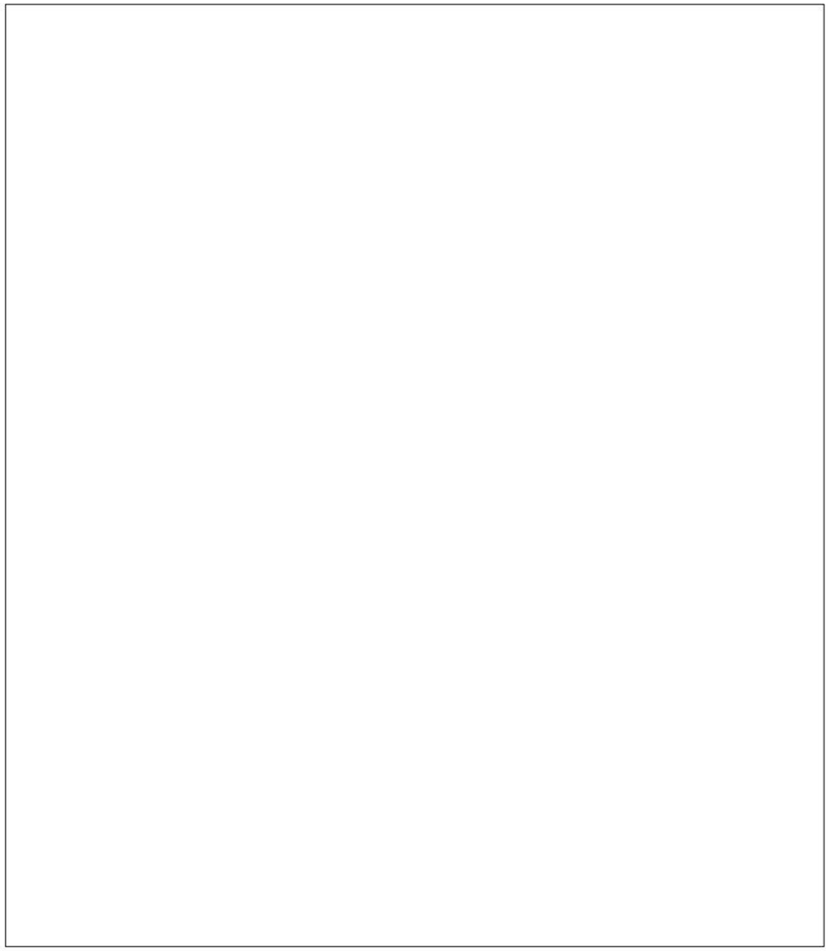
Convierta XML a HTML en Java usando XSLT

En este ejemplo, convertiremos XML a HTML usando el lenguaje XSLT.

2) En segundo lugar, estudiemos qué tipo de resultado (HTML) queremos obtener.

Salida (HTML)

HTML+CSS



Convierta XML a HTML en Java usando XSLT

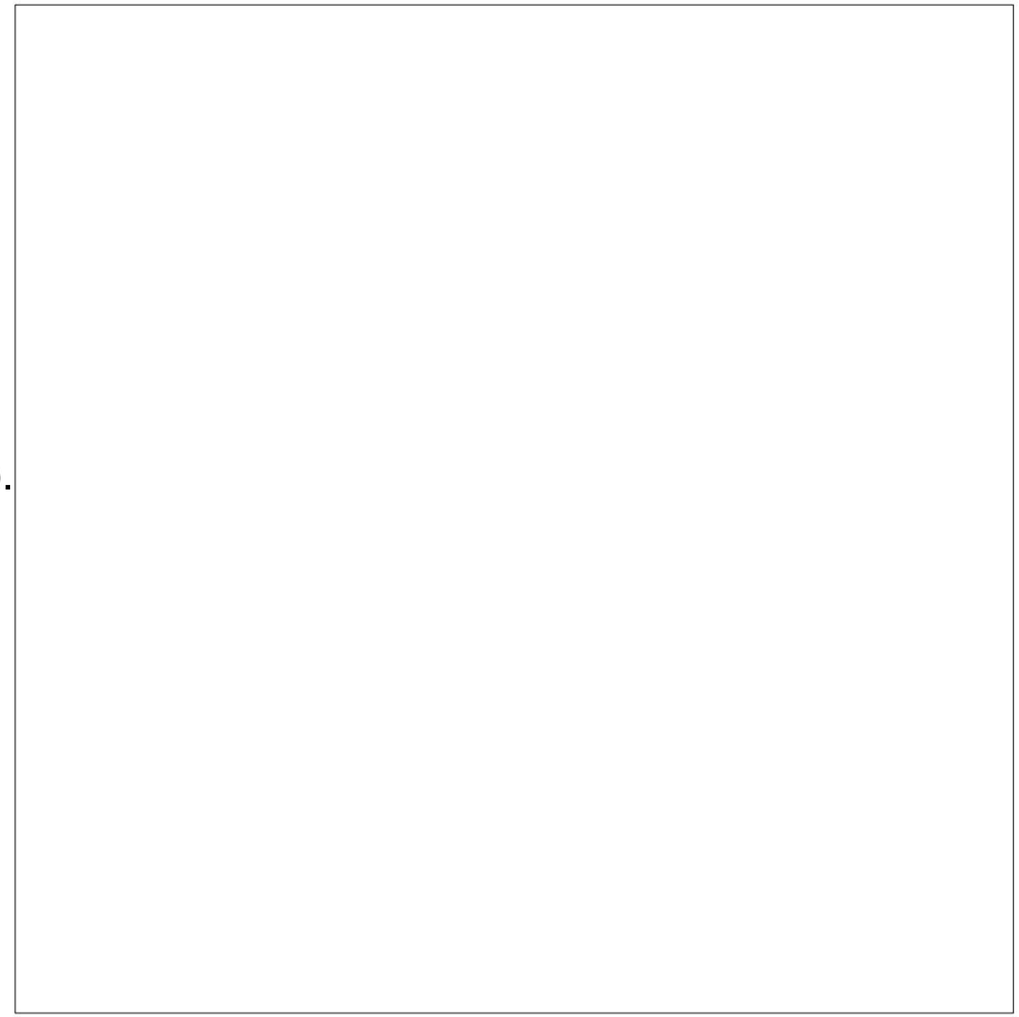
En este ejemplo, convertiremos XML a HTML usando el lenguaje XSLT.

3) En tercer lugar, defina el archivo XSL.

Básicamente, xsl:template match="/" se ejecuta al principio, agregando los encabezados del documento y la tabla.

Luego, xsl:para cada nodo que coincida con los criterios, se agrega una nueva fila (tr) con sus columnas (tr) y datos (xsl:valor-de).

XLS



Convierta XML a HTML en Java usando XSLT

En este ejemplo, convertiremos XML a HTML usando el lenguaje XSLT.

4) Cuarto, aplicar las transformaciones.

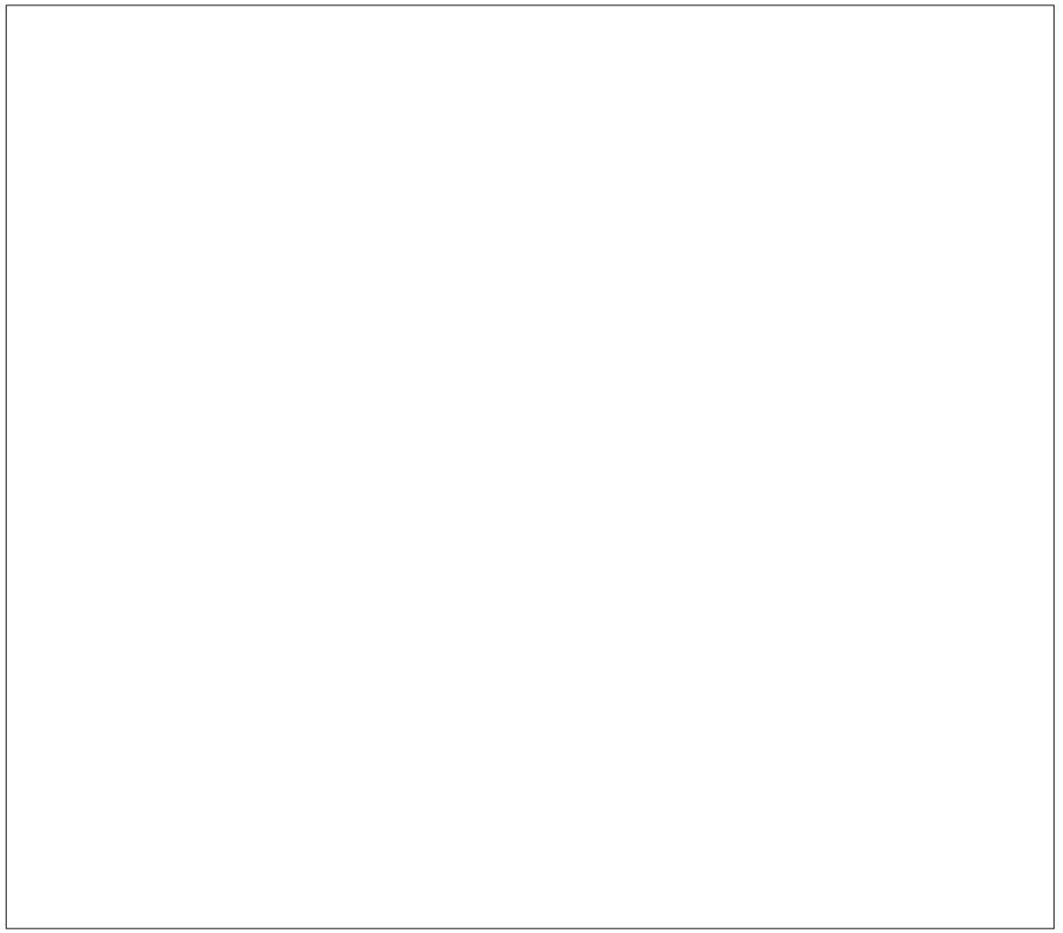
Para transformar el XML en este HTML particular tenemos que unir el XML con el XSL y dejar que Java haga el resto.

Tenemos que crear:

- Un StreamSource para XML
- Un StreamSource para XSL
- Un StringWriter para realizar la transformación.
- Un FileWriter para el HTML resultante

Una instancia de TransformerFactory hará el resto de una forma muy intuitiva.

Ten en cuenta todas las excepciones que tienes que gestionar.



7. ACTIVIDADES PROPUESTAS

Actividades propuestas

Consulta las sugerencias de ejercicios que encontrarás en el “Aula Virtual”. Estas actividades son opcionales y no evaluables, pero comprenderlas es esencial para resolver la tarea evaluable que tenemos por delante.

En breve encontrará las soluciones propuestas adjuntas en un único PDF.

¿Ahora que?

Esta semana deberías…

1) Consulta las soluciones de los ejercicios sugeridos la semana pasada.
2) Estudie este documento y todos los recursos externos que necesite para comprender los contenidos sugeridos para esta semana.
3) Intenta hacer los ejercicios sugeridos. Acude al foro de UNIT para compartir tus dudas e implementaciones alternativas con el resto de alumnos.
4) Revisar los materiales para la próxima semana ANTES de asistir al próximo CT.

8. BIBLIOGRAFÍA

Recursos

- Escuelas W3. Introducción al lenguaje XML. https://www.w3schools.com/xml/
- Abrir llave. Tutorial XML. https://www.abrirllave.com/
- Punto de tutorías. XML-Tutorial. https://www.tutorialspoint.com/es/xml/index.htm
- Documentación de Oracle. SAXÓFONO. https://docs.oracle.com/javase/tutorial/jaxp/sax/index.html
- Documentación de Oracle. DOM. https://docs.oracle.com/javase/tutorial/jaxp/dom/index.html
- Josep Cañellas Bornas, Isidre Guixà Miranda. Acceso a datos. Desarrollo de aplicaciones multiplataforma. Creative Commons. Departamento de Enseñanza, Institut Obert de Catalunya.
  Depósito legal: B. 29430-2013. https://ioc.xtec.cat/educacio/recursos
- Alberto Oliva Molina. Acceso a datos. UD 1. Trabajo con ficheros XML. IES Tubalcaín. Tarazona (Zaragoza, España).

GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura i Esport

ceedcv CENTRE ESPECÍFIC D'EDUCACIÓ A DISTÀNCIA DE LA COMUNITAT VALENCIANA