



# Acceso a datos

---

## Actividad UT 01 Procesamiento de ficheros XML con DOM y SAX

---

### CONTENIDO

1.- Objetivos .....	2
2.- Visión general de la actividad .....	2
3.- Equipos de trabajo .....	2
4.- Requisitos previos .....	2
5.- Evaluación y entrega .....	2
6.- Actividades a realizar .....	3
6.1.- Estudio y análisis del XML proporcionado .....	3
6.2.- Programa para cálculo de valores medios en formato XML.....	3
6.3.- Programa para generación de un informe de valores medios en formato texto.....	4

### 1.- Objetivos

- Recordar y practicar las operaciones con ficheros de texto que se vieron el año pasado en programación.
- Recordar los conceptos relacionados con XML que se vieron el año pasado en lenguajes de marcas: estructura de un XML, declaración, nodos, elementos, atributos, etc.
- Realizar procesamiento de documentos XML usando DOM
- Realizar generación de documentos XML usando DOM
- Realizar procesamiento de documentos XML usando SAX
- Realizar generación de ficheros de texto a partir de datos generados en un procesamiento SAX.

### 2.- Visión general de la actividad

La actividad está enmarcada dentro de la UT 01 (Manejo de ficheros).

En esta actividad se desarrollarán dos programas independientes que permitirán procesar un XML proporcionado, para generar distintas salidas, usando diferentes técnicas.

### 3.- Equipos de trabajo

Esta actividad se realizará en equipos, en principio serán los mismos que se hicieron para la actividad de la UT 02. También se permitirá que los equipos unipersonales se agrupen para formar equipos de dos o tres personas.

En esta actividad todos los equipos deben desarrollar la misma cantidad de tareas.

### 4.- Requisitos previos

- IntelliJ Idea u otro IDE a elección del alumno.

### 5.- Evaluación y entrega

La evaluación se realizará por observación en el aula. Una vez que el equipo considere que la actividad está completada, informará al profesor, que pedirá a los alumnos que realicen una serie de pruebas, y realizará algunas preguntas sobre el código, forma de trabajo, dificultades encontradas o cualquier otro asunto relacionado con la actividad. Se valorarán cosas como

- Ajuste a los requisitos de la actividad. ¿Hace lo que se pedía? ¿Tiene errores?
- Estructura y claridad, uso de convenciones y estándares, elección de nombres de paquetes, clases, variables, métodos, constantes, etc.
- Aplicación correcta de las técnicas, patrones, o arquitecturas requeridas.
- Documentación: JavaDoc y comentarios cuando sean necesarios.

Para poder registrar la calificación, los alumnos entregarán el código fuente de la actividad en formato ZIP. Todos los miembros del equipo deben realizar la entrega del código.

La calificación será la misma para todos los integrantes del grupo, salvo casos evidentes en los que alguno de los integrantes no haya colaborado lo suficiente, en cuyo caso el profesor podrá calificar con menos nota a ese miembro del equipo.

## 6.- Actividades a realizar

### 6.1.- Estudio y análisis del XML proporcionado

Descargar del aula virtual el documento XML weather-data.xml, abrirlo con algún editor adecuado, y analizar su estructura. Identificar los elementos que hacen referencias a ciudades, estaciones meteorológicas, fechas, y periodos de medida. Y también qué mediciones se están registrando (temperatura, humedad, etc.)

A lo largo de esta actividad se hará referencia a la información contenida en el XML, y un estudio previo ayudará a identificar mejor los elementos afectados.

### 6.2.- Programa para cálculo de valores medios en formato XML

Crear un programa Java que, usando DOM, lea el documento weather-data.xml, y genere, también usando DOM, un nuevo documento XML con la siguiente estructura:

```
<city-weather-summary>
  <city id="c001" name="Madrid">
    <average-temperature>
      <avg>17.5</max>
      <max>19.5</max>
      <min>13.2</min>
    </average-temperature>
    <average-humidity>75.3</average-humidity>
    <average-pressure>1014.2</average-pressure>
  </city>
  <!-- Otras ciudades -->
</city-weather-summary>
```

Los datos que aparecen en este ejemplo pueden no ser correctos. No es necesario que se obtengan estos datos exactamente.

En este nuevo documento:

- Aparecerá un elemento “city” por cada ciudad que haya en weather-data.xml.
- En “average-temperature” aparecerán tres valores:
  - “max” – La temperatura máxima de la ciudad. Para calcular el máximo se deben tener en cuenta todas las mediciones de temperatura máxima en todas las estaciones, medidas y periodos que figuran en la ciudad el en XML original.
  - “min” – Igual que la temperatura máxima, pero teniendo en cuenta la temperatura mínima.
  - “avg” – La media de las temperaturas. Para calcular la media de las temperaturas, se realizará con la suma de todas las temperaturas (máxima y mínima) de todas las estaciones, medidas y periodos de la ciudad, dividido del número de temperaturas sumadas. Se redondeará con dos decimales.
- En “average-humidity” aparecerá la media de la humedad de la ciudad, calculada con todas las medidas de humedad de la ciudad, usando todas las estaciones, medidas y periodos. Se redondeará con dos decimales.
- En “average-pressure” aparecerá la media de la presión atmosférica de la ciudad, calculada con todas las medidas de presión de la ciudad, usando todas las estaciones, medidas y periodos. Se redondeará con dos decimales.

Como se ha comentado ya, hay ciertos requisitos para leer el XML original, y para generar el XML de salida.

- Petición de datos de los ficheros de entrada y salida
  - Se preguntará al usuario la ubicación del fichero XML a procesar (entrada)

- Si el fichero no existe, se mostrará un mensaje de error y el programa terminará.
- Si el fichero existe, se continúa la ejecución
- Se preguntará al usuario la ubicación del fichero XML de salida.
  - Si el fichero ya existe
    - Se preguntará al usuario si quiere sobrescribirlo.
    - Si el usuario responde que no, el programa terminará.
    - Si el usuario responde que sí, el programa continuará.
  - Si el fichero no existe, el programa continuará.
- Lectura del XML original:
  - Debe realizarse con DOM.
  - La búsqueda de nodos dentro de cada ciudad puede hacerse con los métodos de los objetos relacionados con DOM, o puede usarse algún lenguaje de consulta como XPath. Se valorará mejor el uso de XPath, ya que implicará investigación por parte del alumno.
- Generación del XML de salida:
  - Debe realizarse con DOM
  - Debe crearse un documento vacío al que se irán añadiendo elementos para conseguir el elemento indicado en el ejemplo.

Para la realización de este programa se pueden seguir dos enfoques distintos:

- Generación de los datos a exportar en alguna clase de estructura (lista de objetos) en memoria, y posterior exportación.
- Generación “al vuelo” del documento de salida, a la vez que se va procesando el documento de entrada.

Cualquiera de los dos enfoques se admite y se valorarán de la misma forma.

### 6.3.- Programa para generación de un informe de valores medios en formato texto.

Crear un programa Java que, usando SAX, lea el documento weather-data.xml, y genere un pequeño informe en formato texto con la siguiente estructura:

Madrid				
-----				
T mínima	T máxima	T media	Humedad media	Presión media
15.22	25.50	18.24	59.68	10.56
Otra ciudad				
-----				
T mínima	T máxima	T media	Humedad media	Presión media
15.22	25.50	18.24	59.68	10.56
Madrid				
-----				
T mínima	T máxima	T media	Humedad media	Presión media
15.22	25.50	18.24	59.68	10.56

Los datos que aparecen en este ejemplo nos son correctos, son inventados. No es necesario que se obtengan estos datos exactamente. El informe debe guardarse en un fichero de texto.

En este nuevo documento, por cada ciudad, aparecerá:

- El nombre de la ciudad, y debajo de este nombre una línea de guiones.

- Bajo la línea de guiones aparecerá una “cabecera” para los cinco valores que se deben calcular. Son los mismos que en el programa del punto 6.2.
- Bajo la línea de cabecera, aparecerán los valores calculados. La forma de calcularlos es la misma que en el apartado 6.2.

Como se ha comentado ya, hay ciertos requisitos para leer el XML original, y para generar el XML de salida.

- Petición de datos de los ficheros de entrada y salida
  - Se preguntará al usuario la ubicación del fichero XML a procesar (entrada)
    - Si el fichero no existe, se mostrará un mensaje de error y el programa terminará.
    - Si el fichero existe, se continúa la ejecución
  - Se preguntará al usuario la ubicación del fichero de texto de salida.
    - Si el fichero ya existe
      - Se preguntará al usuario si quiere sobrescribirlo.
      - Si el usuario responde que no, el programa terminará.
      - Si el usuario responde que sí, el programa continuará.
    - Si el fichero no existe, el programa continuará.
- Lectura del XML original:
  - Debe realizarse con SAX.
- Generación del fichero de salida:
  - Puede realizarse con cualquier clase de manejo de streams que se considere adecuada, pero se recomienda usar PrintWriter.

Para la realización de este programa se pueden seguir dos enfoques distintos:

- Generación de los datos a exportar en alguna clase de estructura (lista de objetos) en memoria, y posterior exportación.
- Generación “al vuelo” del documento de salida, a la vez que se va procesando el documento de entrada con SAX.

Cualquiera de los dos enfoques se admite y se valorarán de la misma forma.