



Java™

Desarrollo de Aplicaciones Web con JEE

- **SAX Y DOM**

1. Manejo de ficheros

1.1 Acceso a fichero

XML (eXtensible Markup Language – Lenguaje de Etiquetado Extensible) es un metalenguaje, es decir, es un lenguaje de etiquetas que se basa en documentos de texto plano en los que se utilizan etiquetas para delimitar los elementos de un documento. Su principal propósito es facilitar el intercambio de información. Son muchos los lenguajes basados en XML (por ejemplo, SVG, RDF, MathML, XSIL, ...). **XML permite a las aplicaciones parsear el contenido del documento**, pudiendo modificar y leer su contenido.

Los ficheros XML se pueden utilizar para proporcionar datos a una base de datos o para almacenar copias de partes del contenido de la base de datos. También se utilizan para escribir ficheros de configuración de programas, para ejecutar comandos en servidores remotos.

Para leer los ficheros XML y acceder a su contenido y estructura, se utiliza un **procesador de XML o parser**. **El procesador lee los documentos y proporciona acceso a su contenido y estructura**. Algunos de los procesadores más empleados son: **DOM**: Modelo de Objetos de Documento y **SAX**: API Simple para XML. Son independientes del lenguaje de programación y existen versiones particulares para Java, VisualBasic, C, etc. Utilizan dos enfoques muy diferentes:

- ✓ **DOM**: un procesador XML que utilice este planteamiento **almacena toda la estructura del documento en memoria en forma de árbol con nodos padre, nodos hijo y nodos finales**. Una vez creado el árbol, se van recorriendo los diferentes nodos y se analiza a qué tipo particular pertenecen. Este tipo de procesamiento necesita más recursos de memoria y tiempo sobre todo si los ficheros XML a procesar son bastante grandes y complejos.
- ✓ **SAX**: un procesador que utilice este planteamiento **lee un fichero XML de forma secuencial y produce una secuencia de eventos** (comienzo/fin del documento, comienzo/fin de una etiqueta, etc.) en función de los resultados de la lectura. Cada evento invoca a un método definido por el programador. Este tipo de procesamiento prácticamente no consume memoria, pero por otra parte, impide tener una visión global del documento por el que navegar.

SAX es menos potente que DOM, pero aun así es más rápido y útil para los casos donde por ejemplo interesa hacer una lectura secuencial o buscar algún fragmento en un documento.