

UA02. PRIMER ACCESO A DATOS

1. Ficheros XML

Probablemente hayas estudiado ya **XML**, bien porque hayas cursado el módulo **Lenguajes de marcas y sistemas de gestión de Información**, o bien porque lo conozcas por tu cuenta. Si no conoces XML, te recomendamos que te familiarices con él, hay múltiples tutoriales y cursos en Internet.

El **metalenguaje XML** se crea para evitar problemas de interoperabilidad entre plataformas y redes. Con él se consigue un soporte estándar para el intercambio de datos: no sólo los datos están en un formato estándar sino también la forma de acceder a ellos. Y entre las ventajas de su uso destacamos:

- Facilita el **Intercambio de Información** entre distintas aplicaciones ya que se basa en estándares aceptados.
- Proporciona una **visión estructurada** de la información, lo que permite su posterior tratamiento de forma local.

Conceptos previos

¿Cómo se trabaja con datos XML desde el punto de vista del desarrollador de aplicaciones?

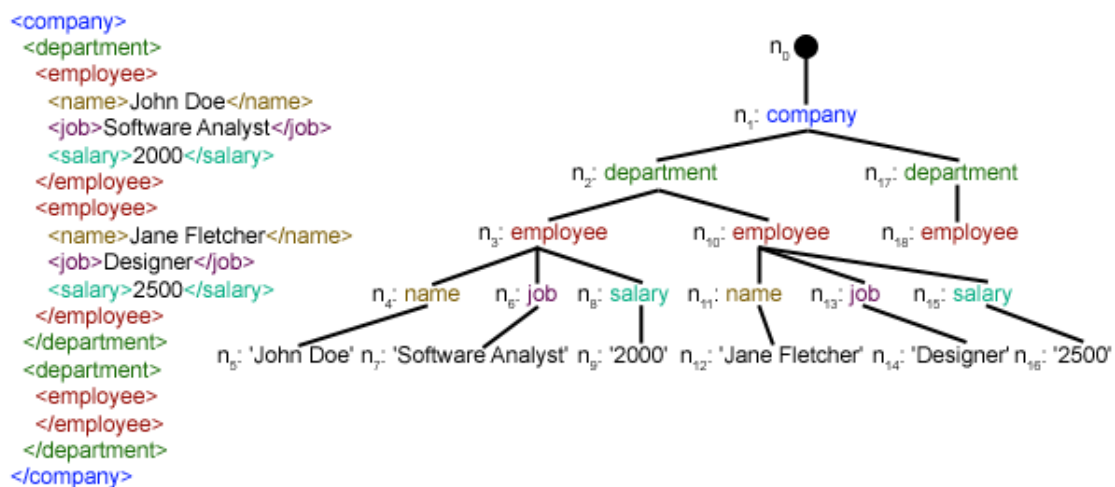
Una aplicación que consume información XML debe:

- Leer un fichero de texto codificado según dicho estándar.
- Cargar la información en memoria y, desde allí...

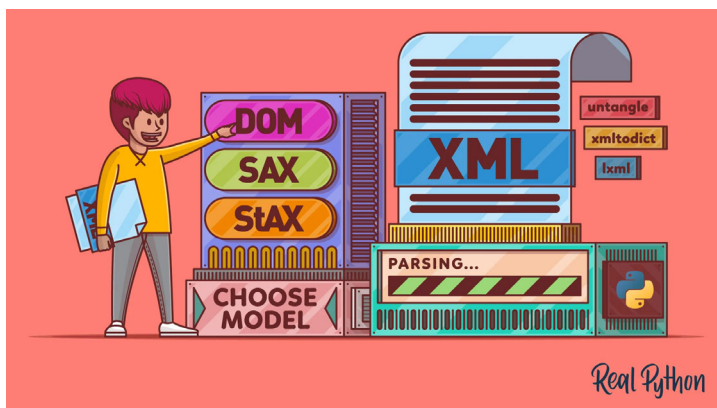
- Procesar esos datos para obtener unos resultados (que posiblemente también almacenará de forma persistente en otro fichero XML).

El proceso anterior se enmarca dentro de lo que se conoce en informática como **"parsing"** o **análisis léxico-sintáctico**

<https://es.wikipedia.org/wiki/Analizador_sint%C3%A1ctico> , y los programas que lo llevan a cabo se denominan **"parsers"** o analizadores (léxico-sintácticos). Más específicamente podemos decir que el **"parsing XML"** es el proceso mediante el cual se lee y se analiza un documento XML para comprobar que está bien formado para, posteriormente, pasar el contenido de ese documento a una aplicación cliente que necesite consumir dicha información.



Algunos de los procesadores más empleados son:



- **DOM (Modelo de Objetos de Documento)**: 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 (que son aquellos que no tienen descendientes). 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 son bastante grandes y complejos.

- **SAX (API Simple para XML)**: 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, etcétera) 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.

Ejemplos

Ejemplo de lectura de fichero XML usando DOM

<<https://javiergarciaescobedo.es/programacion-en-java/53-xml/247-ejemplo-de-lectura-de-fichero-xml-usando-dom>>

Ejemplo de lectura de fichero XML usando SAX <<https://decodigo.com/java-leer-xml-con-sax-parser>>

Obra publicada con **Licencia Creative Commons Reconocimiento Compartir igual 4.0**
<<http://creativecommons.org/licenses/by-sa/4.0/>>