



UD2: Manejo de ficheros

Índice

- Introducción
- Formas de acceso a un fichero
- Operaciones sobre ficheros
- Ficheros binarios
- Ficheros de texto
 - Ficheros XML

Introducción

Un fichero o archivo es un conjunto de bits almacenado en un dispositivo, como por ejemplo un disco duro. La **ventaja** de utilizar ficheros es que los datos que guardamos permanecen en el dispositivo aun cuando apaguemos el ordenador, es decir, **NO son volátiles**. Los ficheros tienen un nombre y se ubican en directorios o carpetas, el nombre debe ser único en ese directorio; es decir, no puede haber dos ficheros con el mismo nombre en el mismo directorio. Por convención cuentan con diferentes extensiones que por lo general suelen ser de 3 letras (PDF, DOC, GIF,...) y nos permiten saber el tipo de archivo.

Formas de acceso a un fichero



Hay dos formas de acceso a la información almacenada en un fichero:

- **Acceso secuencial**: los datos o registros se leen y se escriben en orden. Si se quiere acceder a un dato o un registro que está hacia la mitad del fichero es necesario leer antes todos los anteriores. La escritura de datos se hará a partir del último dato escrito, no es posible hacer inserciones entre los datos que ya hay escritos.
- **Acceso directo o aleatorio**: permite acceder directamente a un dato o registro sin necesidad de leer los anteriores y se puede acceder a la información en cualquier orden. Los datos están almacenados en registros de tamaño conocido, nos podemos mover de un registro a otro de forma aleatoria para leerlos o modificarlos.

Operaciones sobre ficheros



Las operaciones básicas que se realizan sobre cualquier fichero independientemente de la forma de acceso al mismo son las siguientes:

- **Creación del fichero**. El fichero se crea en el disco con un nombre que después se debe utilizar para acceder a él.
- **Apertura del fichero**. Para que un programa pueda operar con un fichero, la primera operación que tiene que realizar es la apertura del mismo.
- **Cierre del fichero***. El fichero se debe cerrar cuando el programa no lo vaya a utilizar.
- **Lectura de los datos del fichero**. Este proceso consiste en transferir información del fichero a la memoria principal, normalmente a través de alguna variable o variables de nuestro programa.
- **Escritura de datos en el fichero**. En este caso el proceso consiste en transferir información de la memoria (por medio de las variables del programa) al fichero.

Normalmente las operaciones típicas que se realizan sobre un fichero una vez abierto son las siguientes:

- **Altas**: consiste en añadir un nuevo registro al fichero.
- **Bajas**: consiste en eliminar del fichero un registro ya existente. La eliminación puede ser lógica, cambiando el valor de algún campo del registro que usemos para controlar dicha situación; o física, eliminando físicamente el registro del fichero.
- **Modificaciones**: consiste en cambiar parte del contenido de un registro.
- **Consultas**: consiste en buscar en el fichero un registro determinado.

Operaciones sobre ficheros secuenciales

En los ficheros secuenciales los registros se insertan en orden cronológico, es decir, un registro se inserta a continuación del último insertado. Veamos como se realizan las operaciones típicas:

- **Consultas**: para consultar un determinado registro es necesario empezar la lectura desde el primer registro, y continuar leyendo secuencialmente hasta localizar el registro buscado.
- **Altas**: en un fichero secuencial las altas se realizan al final del último registro insertado.

Operaciones sobre ficheros secuenciales

- **Bajas**: para dar de baja un registro de un fichero es necesario leer todos los registros uno a uno y escribirlos en un fichero auxiliar, salvo el que deseamos dar de baja. Una vez reescritos hemos de borrar el fichero inicial y renombrar el fichero auxiliar dándole el nombre del fichero original.
- **Modificaciones**: consiste en localizar el registro a modificar, efectuar la modificación y reescribir el fichero inicial en otro fichero auxiliar que incluya el registro modificado. proceso es similar a las bajas.

Operaciones sobre ficheros aleatorios

Las operaciones en ficheros aleatorios son las vistas anteriormente pero teniendo en cuenta que para acceder a un registro hay que localizar la posición o dirección donde se encuentra.

Normalmente para posicionarnos en un registro es necesario aplicar una **función de conversión** que usualmente está relacionada con el tamaño del registro y con la clave del mismo (la clave es el campo o campos que identifica de forma unívoca a un registro). Por ejemplo, disponemos de un fichero de empleados entonces para localizar al empleado con identificador X necesitamos acceder a la posición $\text{tamaño} \times (X-1)$ para acceder a los datos de dicho empleado.

Operaciones sobre ficheros aleatorios

Veamos cómo se realizan las operaciones típicas:

- **Consultas**: para consultar un determinado registro necesitamos saber su clave, aplicar la función de conversión a la clave para obtener la dirección y leer el registro ubicado en esa posición.
- **Altas**: para insertar un registro necesitamos saber su clave, aplicar la función de conversión a la clave para obtener la dirección y escribir el registro en la posición devuelta.
- **Bajas**: las bajas suelen realizarse de forma lógica, es decir, se suele utilizar un campo del registro a modo de switch que tenga el valor 1 cuando el registro exista y le damos el valor 0 para darle de baja.
- **Modificaciones**: para modificar un registro hay que localizarlo, necesitamos saber su clave para aplicar la función de conversión y así obtener la dirección, modificar los datos que nos interesen y reescribir el registro en esa posición.

Ficheros binarios



Ficheros binarios

Los ficheros binarios almacenan secuencias de dígitos binarios que no son legibles directamente por el usuario como ocurría con los ficheros de texto.

Cada tipo de dato ocupa un espacio medido en bytes. El tamaño de dichos tipos varía de un lenguaje de programación a otro. Por ejemplo un entero ocupa 4 bytes en Java y 12 bytes en Python.

Ficheros binarios – Objetos serializables

Por ejemplo, Java y Python nos permiten guardar objetos en ficheros binarios. La serialización de objetos permite tomar cualquier objeto y convertirlo en una secuencia de bits, que puede ser posteriormente restaurada para regenerar el objeto original.

En Java; para poder hacerlo, el objeto tiene que implementar la interfaz **Serializable** que dispone de una serie de métodos con los que podremos guardar y leer objetos en ficheros binarios.

En Python la librería **pickle** implementa los métodos necesarios para la serialización que en Python se conoce como **Pickling**.

Ficheros aleatorios

Los distintos lenguajes de programación disponen de clases y métodos para acceder al contenido de un fichero binario de forma aleatoria (no secuencial) y para posicionarnos en una posición concreta del mismo.

Ficheros de texto



Ficheros de texto

Los ficheros de texto almacenan la información codificada mediante caracteres, en un sistema de codificación legible directamente por el usuario. (ASCII, UTF-8, UTF-16, ...)

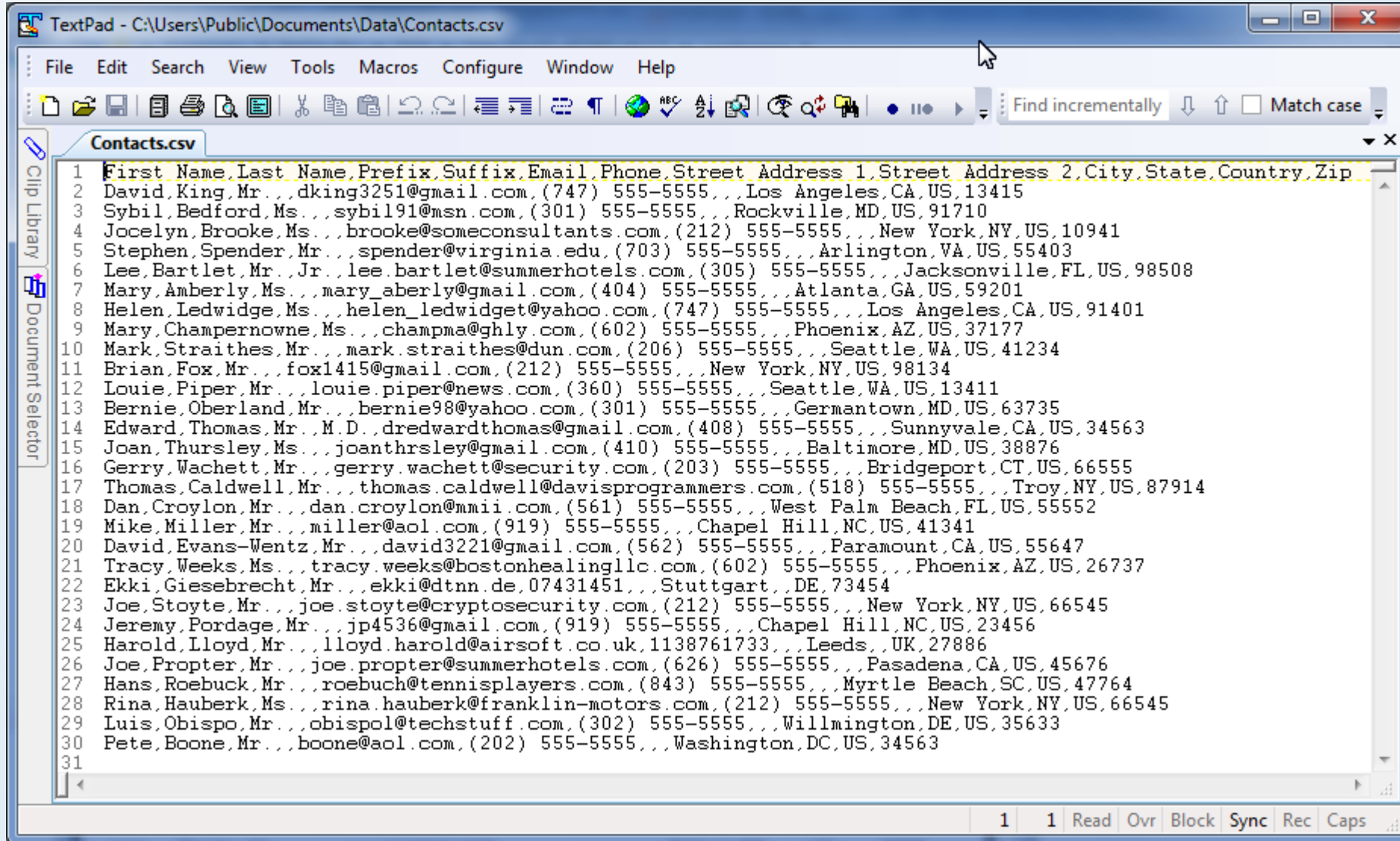
Cada uno de los caracteres ocupa el mismo tamaño (que dependerá del sistema de codificación utilizado, normalmente 8 bits o 1 byte).

Ficheros de texto - Tipos

A día de hoy existen multitud de tipos de ficheros de texto, pero los más significativos son:

- **CSV**: Fichero en el que cada registro se almacena en una línea y los campos dentro del registro se divide por un carácter o conjunto de caracteres conocido (normalmente `,` o `;`) La primera del fichero puede contener información de cabecera de la información que contiene dicho campo.

Ficheros de texto – Tipos (Ejemplo CSV)



The screenshot shows a TextPad window titled 'TextPad - C:\Users\Public\Documents\Data\Contacts.csv'. The menu bar includes File, Edit, Search, View, Tools, Macros, Configure, Window, and Help. The toolbar contains various icons for file operations and editing. The status bar at the bottom shows '1 1 Read Ovr Block Sync Rec Caps'. The main text area displays a CSV file named 'Contacts.csv' with the following content:

```
1 First Name,Last Name,Prefix,Suffix,Email,Phone,Street Address 1,Street Address 2,City,State,Country,Zip
2 David,King,Mr.,,dking3251@gmail.com,(747) 555-5555,,Los Angeles,CA,US,13415
3 Sybil,Bedford,Ms.,,sybil91@msn.com,(301) 555-5555,,Rockville,MD,US,91710
4 Jocelyn,Brooke,Ms.,,brooke@someconsultants.com,(212) 555-5555,,New York,NY,US,10941
5 Stephen,Spender,Mr.,,spender@virginia.edu,(703) 555-5555,,Arlington,VA,US,55403
6 Lee,Bartlet,Mr.,Jr.,lee.bartlet@summerhotels.com,(305) 555-5555,,Jacksonville,FL,US,98508
7 Mary,Amberly,Ms.,,mary_amberly@gmail.com,(404) 555-5555,,Atlanta,GA,US,59201
8 Helen,Ledwidge,Ms.,,helen_ledwidge@yahoo.com,(747) 555-5555,,Los Angeles,CA,US,91401
9 Mary,Champernowne,Ms.,,champma@ghly.com,(602) 555-5555,,Phoenix,AZ,US,37177
10 Mark,Straithes,Mr.,,mark.straites@dun.com,(206) 555-5555,,Seattle,WA,US,41234
11 Brian,Fox,Mr.,,fox1415@gmail.com,(212) 555-5555,,New York,NY,US,98134
12 Louie,Piper,Mr.,,louie.piper@news.com,(360) 555-5555,,Seattle,WA,US,13411
13 Bernie,Oberland,Mr.,,bernie98@yahoo.com,(301) 555-5555,,Germantown,MD,US,63735
14 Edward,Thomas,Mr.,M.D.,dredwardthomas@gmail.com,(408) 555-5555,,Sunnyvale,CA,US,34563
15 Joan,Thursley,Ms.,,joanthrsley@gmail.com,(410) 555-5555,,Baltimore,MD,US,38876
16 Gerry,Wachett,Mr.,,gerry.wachett@security.com,(203) 555-5555,,Bridgeport,CT,US,66555
17 Thomas,Caldwell,Mr.,,thomas.caldwell@davisprogrammers.com,(518) 555-5555,,Troy,NY,US,87914
18 Dan,Croylon,Mr.,,dan.croylon@mmii.com,(561) 555-5555,,West Palm Beach,FL,US,55552
19 Mike,Miller,Mr.,,miller@aol.com,(919) 555-5555,,Chapel Hill,NC,US,41341
20 David,Evans-Wentz,Mr.,,david3221@gmail.com,(562) 555-5555,,Paramount,CA,US,55647
21 Tracy,Weeks,Ms.,,tracy.weeks@bostonhealingllc.com,(602) 555-5555,,Phoenix,AZ,US,26737
22 Ekki,Giesebrecht,Mr.,,ekki@dtm.de,07431451,,Stuttgart,,DE,73454
23 Joe,Stoyte,Mr.,,joe.stoyte@cryptosecurity.com,(212) 555-5555,,New York,NY,US,66545
24 Jeremy,Pordage,Mr.,,jp4536@gmail.com,(919) 555-5555,,Chapel Hill,NC,US,23456
25 Harold,Lloyd,Mr.,,lloyd.harold@airsoft.co.uk,1138761733,,Leeds,,UK,27886
26 Joe,Propter,Mr.,,joe.propter@summerhotels.com,(626) 555-5555,,Pasadena,CA,US,45676
27 Hans,Roebuck,Mr.,,roebuch@tennisplayers.com,(843) 555-5555,,Myrtle Beach,SC,US,47764
28 Rina,Hauberk,Ms.,,rina.hauberk@franklin-motors.com,(212) 555-5555,,New York,NY,US,66545
29 Luis,Obispo,Mr.,,obispol@techstuff.com,(302) 555-5555,,Wilmington,DE,US,35633
30 Pete,Boone,Mr.,,boone@aol.com,(202) 555-5555,,Washington,DC,US,34563
31
```


Ficheros de texto - Tipos

- **XML**: (acrónimo de eXtensible Markup Language, «lenguaje de marcas extensible») es un formato de texto sencillo para el intercambio de datos. Características:
 - La información está encerrada entre etiquetas.
 - Tiene una estructura jerárquica. Hay un elemento raíz y el resto dependen de él.

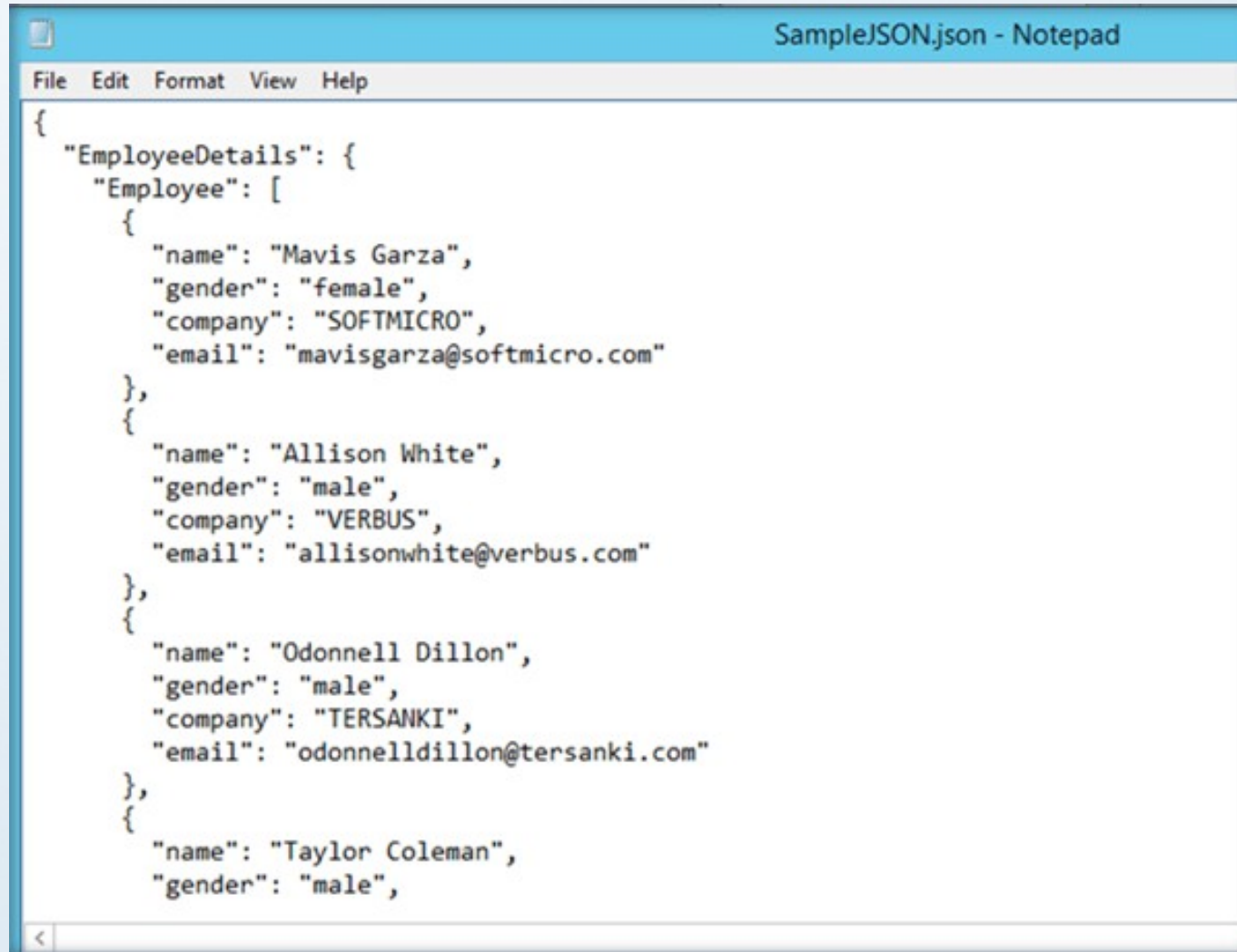
Ficheros de texto – Tipos (Ejemplo XML)

```
<?xml version="1.0" encoding="UTF-8" ?>
- <customer_order number="004985" date="2004-06-24">
- <lines>
- <line no="1">
  <item>Disc CD</item>
  <quantity>30</quantity>
  <price>0.95</price>
</line>
- <line no="2">
  <item>Disc CD-RW</item>
  <quantity>20</quantity>
  <price>2.95</price>
</line>
</lines>
- <customer>
  <name>Technical University of Lublin</name>
  <street>Nadbystrzycka 38</street>
  <city>Lublin</city>
  <post_code>20-501</post_code>
</customer>
- <payment>
  <card_issuer>Master Card</card_issuer>
  <card_number>1234 567890 12345</card_number>
  <expiration_date month="10" year="2005" />
</payment>
</customer_order>
```

Ficheros de texto - Tipos

- **JSON**: (acrónimo de JavaScript Object Notation, «notación de objeto de JavaScript») es un formato de texto sencillo para el intercambio de datos. JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia y cuando la fuente de datos es explícitamente de fiar.
- Si bien se tiende a considerar JSON como una alternativa a XML, lo cierto es que no es infrecuente el uso de JSON y XML en la misma aplicación.

Ficheros de texto – Tipos (Ejemplo JSON)



```
{
  "EmployeeDetails": {
    "Employee": [
      {
        "name": "Mavis Garza",
        "gender": "female",
        "company": "SOFTMICRO",
        "email": "mavisgarza@softmicro.com"
      },
      {
        "name": "Allison White",
        "gender": "male",
        "company": "VERBUS",
        "email": "allisonwhite@verbus.com"
      },
      {
        "name": "Odonnell Dillon",
        "gender": "male",
        "company": "TERSANKI",
        "email": "odonnelldillon@tersanki.com"
      },
      {
        "name": "Taylor Coleman",
        "gender": "male",
```

Ficheros XML



DOM

El Modelo de Objetos del Documento o Document Object Model (DOM) es un modelo de objetos estandarizado para documentos HTML y XML. DOM es un conjunto de interfaces para describir una estructura abstracta para un documento XML.

Los programas que acceden a la estructura de un documento a través de la interfaz de DOM pueden insertarse arbitrariamente, borrarse y reordenar los nodos de un documento XML, esto es, con DOM se puede modificar el contenido, la estructura y el estilo o presentación de los documentos.

DOM carga toda la información del XML en la memoria RAM en una estructura de árbol que nos permite navegarlo como queramos.

SAX

SAX es más complejo de programar que DOM, ya que es necesario crear un parser de XML.

La lectura de un documento XML produce eventos que ocasiona la llamada a métodos, los eventos son encontrar la etiqueta de inicio y fin del documento (`startDocument()` y `endDocument()`), la etiqueta de inicio y fin de un elemento (`startElement()` y `endElement()`), los caracteres entre etiquetas (`characters()`), etc.

SAX está orientado a eventos, es decir, cada vez que el parser analiza un fragmento de XML nos va informando de ello, DOM carga toda la información en memoria del XML en una estructura de árbol que nos permite navegarlo como queramos.

Diferencias SAX y DOM

SAX y DOM son dos mecanismos para parsear documentos XML. Mientras que el primero está orientado a eventos, es decir, cada vez que el parser analiza un fragmento de XML nos va informando de ello, el segundo carga toda la información en memoria del XML en una estructura de árbol que nos permite navegarlo como queramos.

SAX	DOM
Orientado a eventos	Equivalencia XML en clases Java
Carga en memoria una parte del XML	Carga en memoria todo el XML
Adecuado para archivos grandes	Adecuado para archivos pequeños
Se adecua mejor a XML simples	Se adecua mejor a XML complejos

Diferencias SAX y DOM

Siempre que un XML sea excesivamente grande no nos queda más remedio que recurrir a SAX. Cuando el tamaño del XML es razonable, en ese caso podemos escoger cualquiera de los dos escenarios.

Alguna gente alega que DOM es más rápido porque tiene cargada la estructura en memoria, pero obviamente no están teniendo en cuenta el tiempo de generación de esa misma estructura.

Organización de ficheros



Sistema de archivos

El sistema de archivos es el componente del sistema operativo encargado de administrar y facilitar el uso de las memorias periféricas, ya sean secundarias o terciarias. (Discos duros, etc.)

Sus principales funciones son la asignación de espacio a los archivos, la administración del espacio libre y del acceso a los datos resguardados. Estructuran la información guardada en un dispositivo de almacenamiento de datos o unidad de almacenamiento (normalmente un disco duro de una computadora), que luego será representada ya sea textual o gráficamente utilizando un gestor de archivos.

La mayoría de los sistemas operativos manejan su propio sistema de archivos.

Normalmente los archivos almacenados en una unidad de almacenamiento se organiza en una estructura jerárquica. En dicha estructura se utilizan elementos llamados directorios para organizar y agrupar ficheros y directorios.

Las operaciones básicas que se realizan sobre cualquier directorio son las siguientes:

- **Creación del directorio**. El directorio se crea en el disco con un nombre que después se debe utilizar para acceder a él.
- **Listado del directorio**. Se lista el contenido del directorio con los subdirectorios y ficheros contenidos en el mismo.
- **Acceso al directorio**. Permite el acceso al contenido de un directorio. Dicho acceso se puede hacer de forma relativa al directorio en el que nos encontremos ubicados o de forma absoluta al sistema de archivos.

- **Copiar**. Permite hacer una copia de un archivo o directorio.
- **Mover**. Permite mover un archivo o directorio a otro espacio dentro del sistema de archivos.
- **Renombrar**. Permite cambiar el nombre a un archivo o directorio.
- **Eliminar**. Permite eliminar un archivo o directorio del sistema de archivos.