

XML			
DOM		SAX	
Escritura		Lectura	
1 - Creamos una nueva instancia de <b>DocumentBuilderFactory</b> .		1 - Creamos una clase reader que extiende de <b>DefaultHandler</b> .	
2 - Creamos un nuevo <b>DocumentBuilder</b> a <b>dbf</b> .		2 - Override <b>'startDocument()'</b> , <b>'endDocument()'</b> , <b>'startElement()'</b> , <b>'endElement()'</b> y <b>'characters()'</b> .	
3 - Obtenemos la implementación DOM del <b>builder</b> ( <b>DomImplementation</b> ).		3 - En el método <b>'characters()'</b> , hay que crear un nuevo String a partir del char[], para reemplazar todos los saltos de línea y tabulaciones por un espacio.	
4 - Creamos un <b>Document</b> la implementación e introducimos en el medio el <b>Nodo raíz</b> .		4 - Creamos una nueva instancia de <b>SAXParserFactory</b> .	
5 - Establecemos la versión XML al documento creado <b>"1.0"</b>		5 - Creamos un nuevo <b>SAXParser</b> al <b>parserFactory</b> .	
6 - Creamos el XML añadiendo <b>'Element'</b> y <b>'TextNode'</b> con la ayuda de <b>'appendChild'</b> .		6 - Creamos un <b>procesadorXml</b> (XMLReader) obteniendolo del parser usando el <b>'getXMLReader()'</b>	
7 - Creamos un nuevo <b>origen</b> ( <b>DomSource</b> ) y le pasamos por parámetro el <b>Document</b> .		7 - Establecemos al procesadorXML un <b>ContentHandler</b> y le pasamos un objeto de la clase que hemos creado.	
8 - Creamos un <b>resultado</b> ( <b>StreamResult</b> ) y le pasamos por parámetro el <b>File</b> o <b>System.out</b> .		8 - Creamos un nuevo <b>InputSource</b> con la ruta del fichero que queremos leer.	
9 - Creamos un <b>transformador</b> , <b>TransformerFactory</b> nueva instancia, nuevo <b>Transformer</b> .		9 - Por último, llamamos al método <b>'parse'</b> del procesadorxml y le pasamos el <b>InputSource</b> creado.	
10 - Llamamos al método <b>'transform'</b> del transformador y le pasamos el origen y el resultado.			
Lectura		FICHEROS DE TEXTO	
El paso 1 y 2 igual que en la escritura.		FileReader	
3 - Creamos un <b>Document</b> y usamos el builder con el método <b>'parse'</b> al que le pasamos el <b>File</b> .		BufferedReader	
4 - Normalizamos el <b>DocumentElement</b> y llamando al método <b>'normalize()'</b> .		FileWriter	
5 - Mostramos el <b>DocumentElement</b> llamando al método <b>'getNodeName'</b> .		PrintWriter	
6 - Leemos el documento usando:		FICHEROS BINARIOS/OBJETOS	
- <b>NodeList</b> y el método <b>'getElementsByTagName'</b> para los nodos que se repiten.		FileInputStream	
- Para recorrer el <b>NodeList</b> creamos <b>Node</b> y usamos el método <b>'item'</b> .		FileOutputStream	
- Controla que el nodo sea <b>'Node.ELEMENT.NODE'</b> con <b>'getNodeType()'</b> .		DataInputStream	
- Si es así creamos un <b>Element</b> haciendo un parse (Element) al <b>Node</b> .		DataOutputStream	
- Usamos <b>'getElementsByTagName'</b> , obtenemos el primer item y <b>'getTextContent'</b>		ObjectInputStream	
		ObjectOutputStream	
XStream			
Escritura		Lectura	
1 - Creamos el <b>File</b> del fichero que queremos leer y hacemos la lectura normal almacenando los objetos.		1 - En primer lugar creamos la estructura de clases deseada observando el xml dado.	
2 - Creamos clase <b>wrapper</b> de las listas, con el constructor y los metodos <b>'add'</b> , <b>getter</b> , <b>setter</b> y <b>mostrar</b> .		2 - Creamos un nuevo <b>XStream</b> y le pasamos por parámetro un nuevo <b>DomDriver</b> ("UTF-8").	
3 - Creamos un nuevo <b>XStream</b> y le pasamos por parámetro un nuevo <b>DomDriver</b> ("UTF-8").		3 - Disminuimos la seguridad con el método <b>'addPermission'</b> y el <b>'AnyTypePermission.Any'</b> .	
4 - Cambiamos el nombre de las etiquetas XML con el método <b>'alias'</b> .		4 - Cambiamos el nombre de las etiquetas XML con el método <b>'alias'</b> .	
5 - Omitimos algún campo si es necesario con el método <b>'omitField'</b> .		5 - Omitimos algún campo si es necesario con el método <b>'omitField'</b> .	
6 - No olvidar realizar el <b>'addImplicitCollection'</b> a las listas y etiquetas que queramos quitar.		6 - No olvidar realizar el <b>'addImplicitCollection'</b> a las listas y etiquetas que queramos quitar.	
7 - Finalmente usamos el método <b>'toXML'</b> pasando por parámetro la lista creada de objetos anteriormente y un nuevo <b>FileOutputStream</b> con el nombre del fichero xml que queremos crear.		7 - Creamos un objeto y lo recuperamos haciendo parse y con el método <b>'fromXML'</b> pasándole por parámetro un nuevo <b>FileInputStream</b> con el nombre del fichero xml que queremos leer.	
JSON			
SERIALIZACIÓN OBJETO A JSON			
1 - Creamos un nuevo <b>Gson</b> y usamos el método <b>'toJson'</b> para mostrarlo o escribirlo.			
2 - <b>FORMAT</b> : Creamos un <b>Gson</b> , nuevo <b>'GsonBuilder'</b> , con el método <b>'setPrettyPrinting'</b> y el método <b>'create'</b> , finalmente usamos el <b>'toJson'</b> . (También sirve con listas).			
3 - Si queremos que aparezca el nombre del objeto en el JSON debemos usar el <b>'@SerializedName'</b> encima del atributo.			
DESERIALIZAR JSON A OBJETO			
1 - Creamos la clase correspondiente.			
2 - Creamos un nuevo <b>Gson</b> y usamos el método <b>'fromJson'</b> pasandole el String y la clase creada.			
DESERIALIZAR VARIOS OBJETOS A JSON			
1 - Creamos un nuevo <b>Gson</b> .			
2 - Creamos un <b>Type</b> , nuevo <b>TypeToken</b> y le pasamos entre diples el <b>List&lt;Objeto&gt;</b> , paréntesis y corchetes de apertura y cierre, después usamos el <b>'getType'</b> .			
3 - Creamos la lista del objeto correspondiente donde almacenaremos los objetos, usamos el <b>'fromJson'</b> pasándole el String y el <b>Type</b> .			
LECTURA FICHERO JSON			
1 - Creamos un nuevo <b>Gson</b> .		<b>IMPORTANTE:</b> El Gson se tiene que encontrar la estructura de clases tal cual la usa en el fichero Json	
2 - Creamos un <b>InputStream</b> nuevo <b>FileInputStream</b> con el nombre del fichero que queremos leer.			
3 - Creamos un nuevo <b>JsonReader</b> pasándole un nuevo <b>InputStreamReader</b> al que le pasamos el fis y el 'UTF-8'.			
4 - El reader es un <b>flujo que hay que cerrar al final</b> y funciona como un iterador, por lo que podemos meterlo en un while con el <b>'hasNext()'</b> .			
5 - Usamos el método <b>'beginArray'</b> y <b>'endArray'</b> si existe algún array.			
6 - Creamos un objeto del tipo que sea según el Json y lo recuperamos usando el <b>Gson</b> con el método <b>'fromJson'</b> pasándole el <b>reader</b> y la clase a lo que pertenece.			
ESCRITURA FICHERO JSON			
1 - Creamos un nuevo <b>JsonWriter</b> pasándole un nuevo <b>FileWriter</b> con el nombre del fichero <b>.json</b> que queremos crear. (Es un flujo que hay que cerrar al final).			
2 - Utilizamos los métodos <b>'beginArray'</b> , <b>'endArray'</b> , <b>'beginObject'</b> , <b>'endObject'</b> .			
3 - Para los atributos usamos el <b>'name'</b> especificando el nombre que deseamos que aparezca en el <b>.json</b> y el <b>'value'</b> pásánde el valor.			
<b>IMPORTANTE:</b> Para proporcionar un nombre a un objeto/array debe estar envuelto previamente en otro objeto, para poder establecer el nombre antes de abrir el objeto en cuestión.			