



Recorriendo XML con DOM

ÍNDICE DE CONTENIDOS

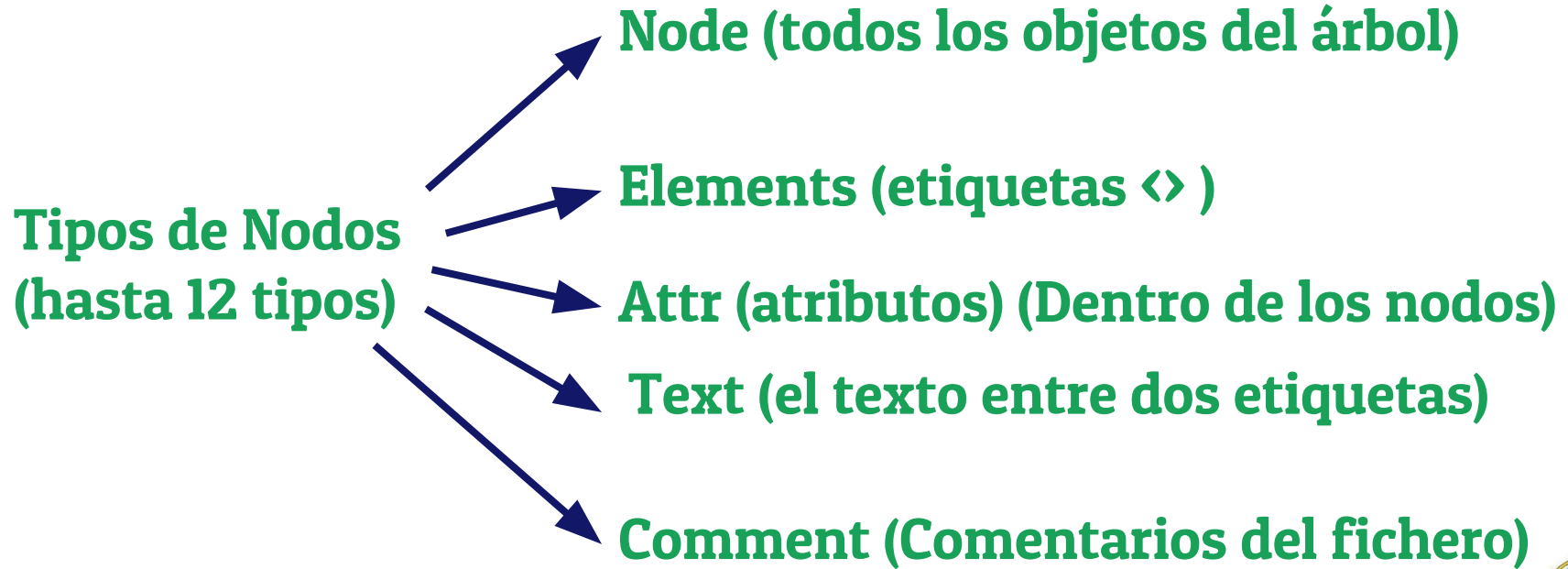
- Características DOM
- Ventajas sobre Sax
- Desventajas sobre Sax
- Ejemplo (Recorrido)

CARACTERÍSTICAS

- Carga en Memoria **TODO** el documento **XML**.
- Estructura de árbol de Nodos.
- Recorrido **y** navegación del árbol creado.



CARACTERÍSTICAS



VENTAJAS SOBRE SAX

- **Permite acceso directo a los elementos del documento.**
 - **Permite crear y modificar documentos.**
- Sax NO**



DESVENTAJA SOBRE SAX

- Limitado por la cantidad de memoria



EJEMPLO

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<personas>
  <persona id="1">
    <nombre>Pepe</nombre>
    <edad>14</edad>
  </persona>
  <persona id="2">
    <nombre>María</nombre>
    <edad>51</edad>
  </persona>
  <persona id="3">
    <nombre>Raul</nombre>
    <edad>22</edad>
  </persona>
</personas>
```



EJEMPLO

1er Paso - Obtenemos el Parser/Document Builder

```
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();  
DocumentBuilder db = dbf.newDocumentBuilder();
```

Posteriormente tenemos muchos métodos para configurar el objeto dbf (formas de validación, espacio de nombres etc...)



EJEMPLO

Parsing del Documento XML(carga en memoria)

Document doc = db.parse(new File("personasXML.xml"));

Esta instrucción carga toda la estructura de datos del fichero XML en memoria

```
args = {String[0]@1016}
dbf = {DocumentBuilderFactoryImpl@1017}
db = {DocumentBuilderImpl@1018}
doc = {DeferredDocumentImpl@1019} "[#document: null]"
  fNodeCount = 30
  fNodeType = {int[32][]@1045}
  fNodeName = {Object[32][]@1047}
  fNodeValue = {Object[32][]@1049}
  fNodeParent = {int[32][]@1050}
  fNodeLastChild = {int[32][]@1051}
  fNodePrevSib = {int[32][]@1052}
  fNodeURI = {Object[32][]@1053}
  fNodeExtra = {int[32][]@1054}
  fidCount = 0
  fidName = null
  fidElement = null
  fNamespacesEnabled = false
```



EJEMPLO

Funciones útiles para trabajar con el documento

//Obtener el Element (etiqueta) raíz

Element root = doc.getDocumentElement();

//Obtener una lista de nodos (de TODOS los TIPOS)

NodeList nl = doc.getElementsByTagName("persona");

//Obtener todos los hijos de un Element

NodeList nl = root.getChildNodes();



EJEMPLO

Funciones útiles para trabajar con el documento

//Obtener el valor del atributo de una Element

String id = root.getAttribute(String s);

//Obtiene el nombre del propio Element

String name = root.getTagName();

//Obtener el tipo de Nodo

nodo.getNodeType();



EJEMPLO

Funciones útiles para trabajar con el documento

//Obtener un nodo de una lista

Node nodo = nl.item(i);

//Comprobar si un Element tiene un atributo

Boolean b = root.hasAttribute("id");

//Obtener el texto de un nodo Texto

String contenido = nt.getWholeText();



EJEMPLO

Funciones útiles para trabajar con el documento

//Obtener todo el texto entre apertura y cierre de un element

String content = e.getTextContent();

//Obtener nombre y valor de un tributos

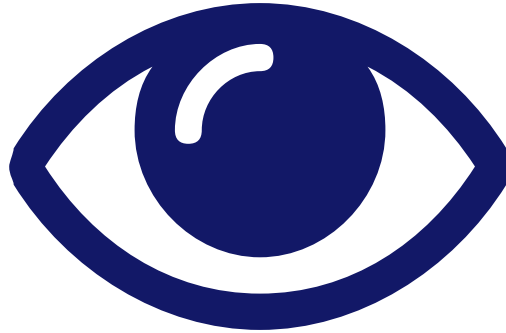
atributo.getName() / atributo.getValue();

//Obtener todo el texto de un nodo (incluyendo hijos)

nodo.getTextContent();



EJEMPLO



END



prof.jdperez@iesalixar.org

