

Recorriendo XML con DOM

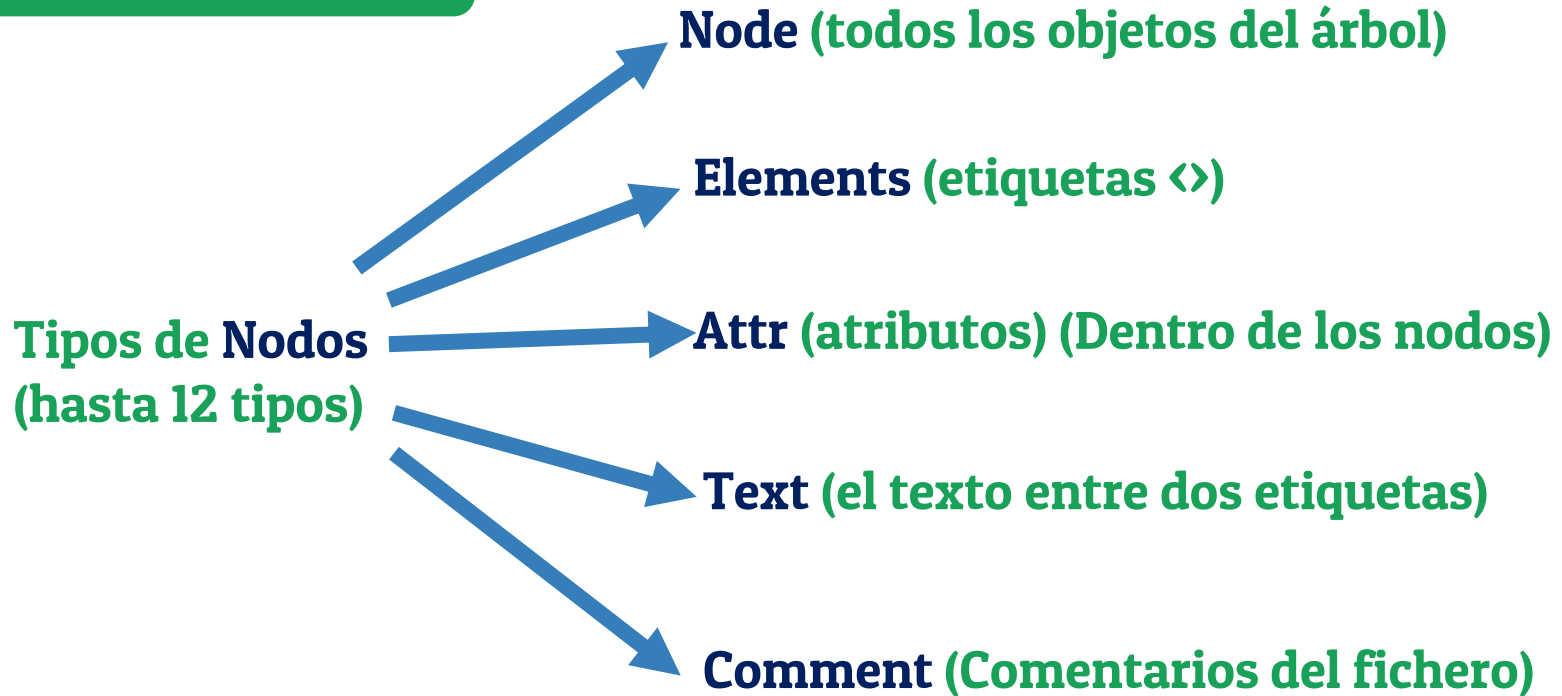
Índice

- **Características DOM**
- **Ventajas/Desventajas sobre SAX**
- **Ejemplo**

Características

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

Características



Ventajas sobre SAX

Ventajas:

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



Desventajas:

- **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

- 2º Paso – Parsing del documentl 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

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);
```

// Obtener 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 atributo

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

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

```
nodo.getTextContent();
```

Fin



jgardur081@g.educaand.es