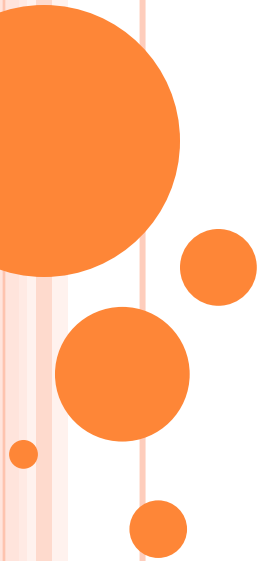

XML en Android

Acceso a datos

http://www.ibm.com/developerworks/library/x-androidxml/

http://developer.android.com/training/basics/network-ops/xml.html

XML

- Android ofrece tres formas de procesar archivos XML:
  - **SAX**, permite procesar los archivos XML nodo a nodo usando disparadores de eventos, sin llegar a procesar el documento XML de forma integral
  - **XmlPullParser**, permite procesar los archivos XML nodo a nodo de forma secuencial, sin llegar a procesar el documento XML de forma integral
  - **DOM**, procesa los documentos XML integralmente, creando la estructura de árbol en memoria

- Todos los métodos están pensados más bien para procesar archivos o datos XML existentes.

Crear un archivo XML I

- Usando el enfoque DOM se puede crear un archivo en formato XML.

```
DocumentBuilderFactory factoria =
        DocumentBuilderFactory.newInstance();
DocumentBuilder constructor =
        factoria.newDocumentBuilder();
DOMImplementation implementacion =
        constructor.getDOMImplementation();
Document documento = implementacion.
         createDocument(null,"nodoraiz", null);
documento.setXmlVersion("1.0");
```

- Obtenemos:

```
<?xml version="1.0" encoding="UTF-8"?>
<nodoraiz>
</nodoraiz>
```

Crear un archivo XML II

- Creación de nodos

```
Element nodo =
        documento.createElement("nodo");
Text texto =
        documento.createTextNode("texto");
nodo.appendChild(texto);
nodo.setAttribute("atributo", "valor");
```

- Obtenemos:

```
<nodo atributo="valor">texto</nodo>
```

Crear un archivo XML III

- Agregar el nodo creado al final del documento XML

```
documento.getDocumentElement().appendChild(
    nodo);
```

- Obtenemos:

```
<?xml version="1.0" encoding="UTF-8"?>
<nodoraiz>
    <nodo atributo="valor">texto</nodo>
</nodoraiz>
```

Crear un archivo XML IV

- Guardar el documento XML generado en un archivo.

```
Source fuente = new DOMSource(documento);
Result resultado = new StreamResult(
        new File(getFilesDir(), "archivo.xml"));
Transformer transformador =
        TransformerFactory.newInstance()
        .newTransformer();
transformador.transform(fuente, resultado);
```

Crear archivo con XmlSerializer I

- Preparamos el archivo

```
FileOutputStream fosxml = new FileOutputStream(
  new File(getFilesDir(),"archivo.xml"));
```

- Preparamos el documento XML

```
XmlSerializer docxml = Xml.newSerializer();
docxml.setOutput(fosxml, "UTF-8");
docxml.startDocument(null, Boolean.valueOf(true));
docxml.setFeature("http://xmlpull.org/v1/doc/features.html#indent-output", true);
```

Crear archivo con XmlSerializer II

- Creamos las etiquetas y cerramos el documento

```
docxml.startTag(null, "nodoraiz");
docxml.startTag(null, "nodo");
docxml.attribute(null, "atributo", "valor");
docxml.text("texto");
docxml.endTag(null, "nodo");
docxml.endDocument();
docxml.flush();
fosxml.close();
```

Leer archivo con XmlPullParser I

- Esta clase permite realizar una lectura secuencial del archivo XML. En la variable evento vamos obteniendo el tipo de etiqueta.

```
XmlPullParser lectorxml = Xml.newPullParser();
lectorxml.setInput(new FileInputStream(new File(
    getFilesDir(),"archivo.xml")),"utf-8");
int evento = lectorxml.getEventType();
```

Leer archivo con XmlPullParser II

```
while (evento != XmlPullParser.END_DOCUMENT){
    if(evento == XmlPullParser.START_TAG){
        String etiqueta = lectorxml.getName();
        if(etiqueta.compareTo("nodo")==0){
            atrib = lectorxml.getAttributeValue(null,
                                                "atributo");
            texto = lectorxml.nextText();
        }
    }
}
evento = lectorxml.next();
}
```