

**UNIVERSIDAD EAFIT**

**INGENIERÍA DE SISTEMAS**

**ST0263: Tópicos Especiales en Telemática, grupo 031, 2010-1, Profesor: Edwin Montoya**

**Reto No 5 – Diseño e Implementación de una Biblioteca Digital (Yet Another Digital Library – YADL) con interfaz Web utilizando XML como mecanismo de persistencia de metadatos.**

**Descripción:**

Una Biblioteca Digital (BiDi), es un sistema de información que permite gestionar objetos o documentos digitales. Estos documentos digitales están representados por un conjunto de metadatos descriptivos (titulo, autores, temas, etc), que describen cada uno de los recursos digitales. Estos objetos digitales están almacenados en un Servidor Web tipo Apache.

El sistema tiene 2 subsistemas: 1) subsistema de gestión y 2) subsistema de consulta.

- 1) Subsistema de gestión: Permite a un administrador del sistema: crear, modificar, borrar o listar documentos digitales. Se permite realizar las anteriores operaciones para los metadatos, y respecto al recurso digital en sí mismo, se puede cargar (upload) al servidor web, y debe definirse el comportamiento del recurso digital para las operaciones de modificación y borrado.
- 2) Subsistema de consulta: Permite a cualquier usuario a través de la web, realizar consultas en el sistema. Se utilizará 5 criterios de búsqueda: 1) por todos los campos, 2) por titulo, 3) por autor, 4) por tema, o 5) por identificador. Una vez definido el criterio y digitado el termino, el sistema listará el conjunto de documentos digitales que cumplen dicho criterio.

Un prototipo de la aplicación se puede acceder en: <http://dis.eafit.edu.co/yadl>

**Persistencia de datos:**

- Para almacenar los metadatos, se utilizará un archivo XML (yadl.xml), en el cual se almacenará cada recurso digital. Se debería utilizar una librería para manipulación de documentos XML a través de DOM o SAX, se recomienda si se va a trabajar con Java, utilizar JDOM. Para los grupos de 3 estudiantes, se debe utilizar una base de datos relacional (ej: mysql) o una base de datos XML para hacer persistencia de los datos en XML, se podría utilizar un motor como eXist (exist.sourceforge.net) para realizar la persistencia, búsqueda y recuperación de metadatos.
- Para almacenar los datos (recursos digitales), se utilizará un servidor web, tipo apache.

**Requerimientos:**

- Realizar el diseño e implementación de una aplicación web, que permita implementar esta biblioteca digital.

- Puede utilizar cualquier tecnología Web, de preferencia: Java, .NET, PHP, Ruby, etc
- Utilizar un servidor Apache para almacenar los datos (recursos digitales)
- Utilizar un archivo XML plano, para almacenar los metadatos de YADL
- Grupos de 3 estudiantes: utilizar motores relacionales como mysql o xml como eXist para la persistencia de los metadatos.

A continuación se presenta un archivo yadl.xml con algunos registros ejemplo:

```
<!--yadl.xml -->
<?xml version="1.0" encoding="UTF-8"?>
<yadl>
  <record>
    <title>Java</title>
    <creator>autor1</creator>
    <creator>autor2</creator>
    <subject>programacion</subject>
    <subject>sistemas</subject>
    <subject>lenguajes</subject>
    <identifier>doi:1</identifier>
    <url>http://servidor/yadl/data/doc1.pdf</url>
  </record>
  <record>
    <title>C#</title>
    <creator>autor2</creator>
    <creator>autor3</creator>
    <subject>programacion</subject>
    <subject>microsoft</subject>
    <identifier>doi:2</identifier>
    <url>http://servidor/yadl/data/doc2.pdf</url>
  </record>
  <record>
    <title>Windows 2003</title>
    <creator>autor4</creator>
    <subject>sistemas operativos</subject>
    <identifier>doi:3</identifier>
    <url>http://servidor/yadl/data/doc3.pdf</url>
  </record>
</yadl>
```

A continuación se presenta el esquema (XML Scheme), que define el modelo de persistencia de metadatos en XML:

```
<!--yadl.xsd-->
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="yadl">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="record"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="record">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="title"/>
      <xs:element maxOccurs="unbounded" ref="creator"/>
      <xs:element maxOccurs="unbounded" ref="subject"/>
      <xs:element ref="identifier"/>
      <xs:element ref="url"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="title" type="xs:string"/>
<xs:element name="creator" type="xs:NCName"/>
<xs:element name="subject" type="xs:string"/>
<xs:element name="identifier" type="xs:NMTOKEN"/>
<xs:element name="url" type="xs:anyURI"/>
</xs:schema>

```

DTD (XML Document Type Definition - DTD), que define el modelo de persistencia de metadatos en XML:

```

<!--yadl.dtd-->
<?xml encoding="UTF-8"?>
<!ELEMENT yadl (record)+>
<!ATTLIST yadl
  xmlns CDATA #FIXED ">
<!ELEMENT record (title,creator+,subject+,identifier,url)>
<!ATTLIST record
  xmlns CDATA #FIXED ">
<!ELEMENT title (#PCDATA)>
<!ATTLIST title
  xmlns CDATA #FIXED ">
<!ELEMENT creator (#PCDATA)>
<!ATTLIST creator
  xmlns CDATA #FIXED ">
<!ELEMENT subject (#PCDATA)>
<!ATTLIST subject
  xmlns CDATA #FIXED ">
<!ELEMENT identifier (#PCDATA)>
<!ATTLIST identifier
  xmlns CDATA #FIXED ">
<!ELEMENT url (#PCDATA)>
<!ATTLIST url
  xmlns CDATA #FIXED ">

```

