Java y el XML

M. D. López De Luise

Objetivo

Presentar XML y las herramientas de Java para su manejo

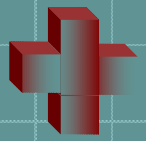
Temario

- Por qué integrar XML con Java.
- Algo de sintaxis XML.
- Lectura/escritura de XML con Java.
- Serialización XML.
- Protocolos XML: XML-RPC y SOAP

M. D. López De Luise

Por qué Java + XML?

datos portables



código portable

I. XML

def XML — lenguaje del tipo markup para intercambio estandarizado de datos

características

- Fácil intercambio
- Personalizable y adaptable
- Archivos son autodocumentados
- Sintaxis describe sencillamente estructuras complejas
- Solo se dedica a describir datos y su estructura
- No describe la presentación de datos (Vs html)
- Separa estructura de la presentación de datos
- Permite describir datos en un archivo autocontenido
- Se apoya en Stylesheet (XSL) para convertirse en html
- Se apoya en DTD
- Se apoya en XML schemas

M. D. López De Luise

I.a.Elementos

<nomtag> texto </nomtag>

<nomtag> elemento </nomtag> → subelemento

<nomtag> texto elemento </nomtag>

<nomtag> elemento texto </nomtag>

start - tag ... end - tag } markups

balanceados

no predefinidos como en html

Root element:

Si engloba a todos los demás

Elemento vacío:

<nomtag/>

<nomtag></nomtag>

I.b.Atributos

- Son propiedades de los elementos a los que acompañan
- Son pares nombre-valor
- Son únicos dentro del mismo tag Vs subelementos (no necesariamente)

formatos posibles

```
<nonTag atributo=valor>
</nomTag>
```

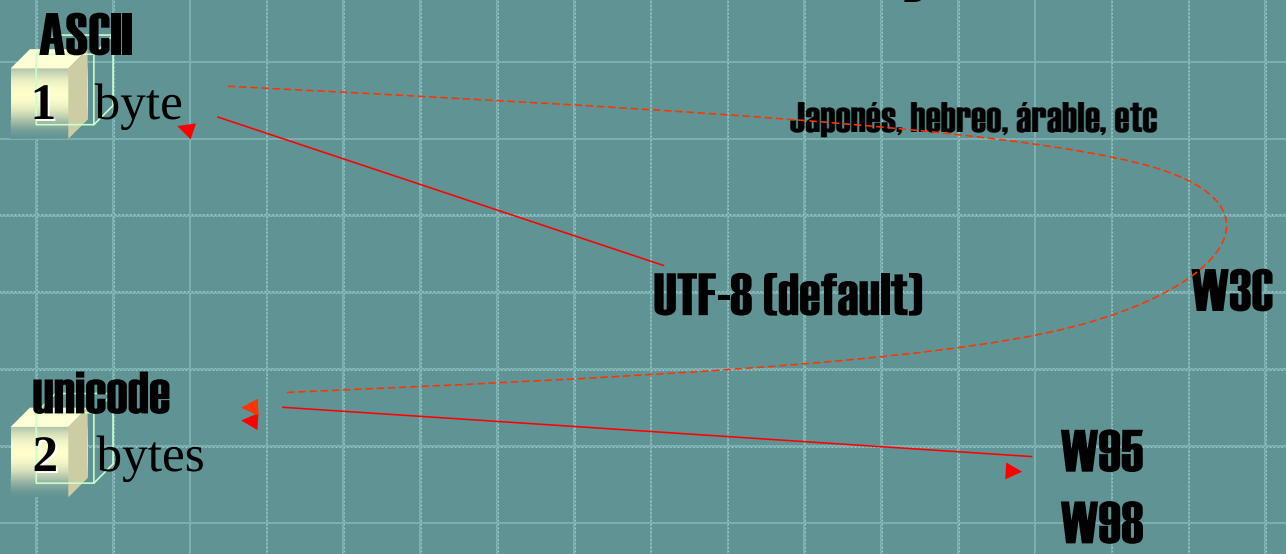
```
<nonTag>
<atributo>valor</atributo>
</nomTag>
```


M. D. López De Luise

I.c.Codificación de los caracteres

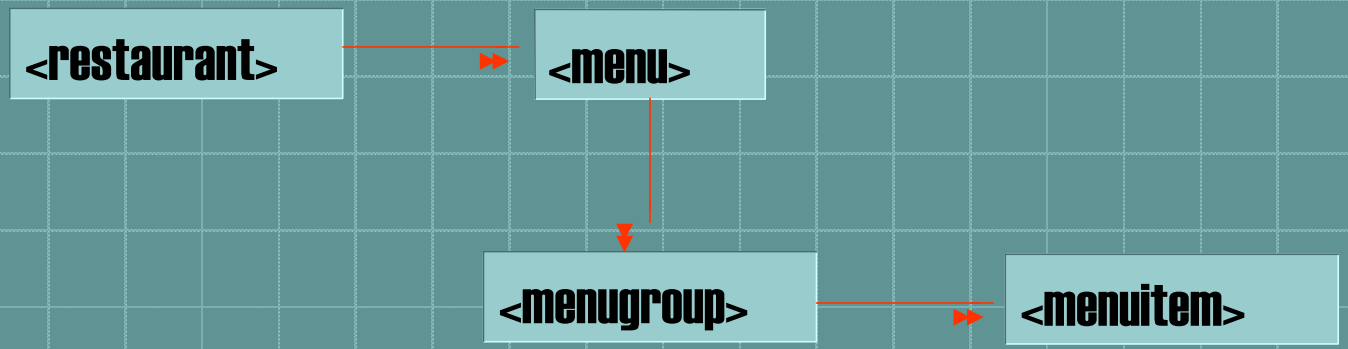
<?xml version="1.0" encoding="UTF-8"?>

(American Standard Code for Information Interchange)



XML

Un ejemplo:



EJEMPLO stylesheet.xsl

Un XSL stylesheet: es un documento XML

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
                xmlns="..." >
  .
  .
  .
  <xsl:template match="pattern">
    template
  </xsl:template>
  .
  .
  .
</xsl:stylesheet>
```

namespace usado para reconocer los elementos XSL

una regla tipo template

otros elementos

1.d.XML bien formado y válido

bien formado

- Tag anidados
- Tag balanceados
- Atributos únicos

válido

- Reglas de la gramática elegida (MathML, Scalable Vector Graphics...)
- Unifica intercambio de datos con terceros

(DTD) Document Type Definition

(XSD) XML Schema Definition

tij_menu.xml

I.e.Namespaces (XSD)

Resuelve colisiones de un nombre de tag con uno definido por la gramática

Define un conj. de nombres y describe los { tipos / elementos / atributos } permitidos en su dominio

Un tipo de namespace frecuente es el XML Schema (cuando está dentro de un XML).

```
<oficina
    xmlns:xsi="http://www.daniela.com/2005/XMLSchema-instance"
    xsi: noNamespaceSchemaLocation="tij_menu.xsd">
```

url

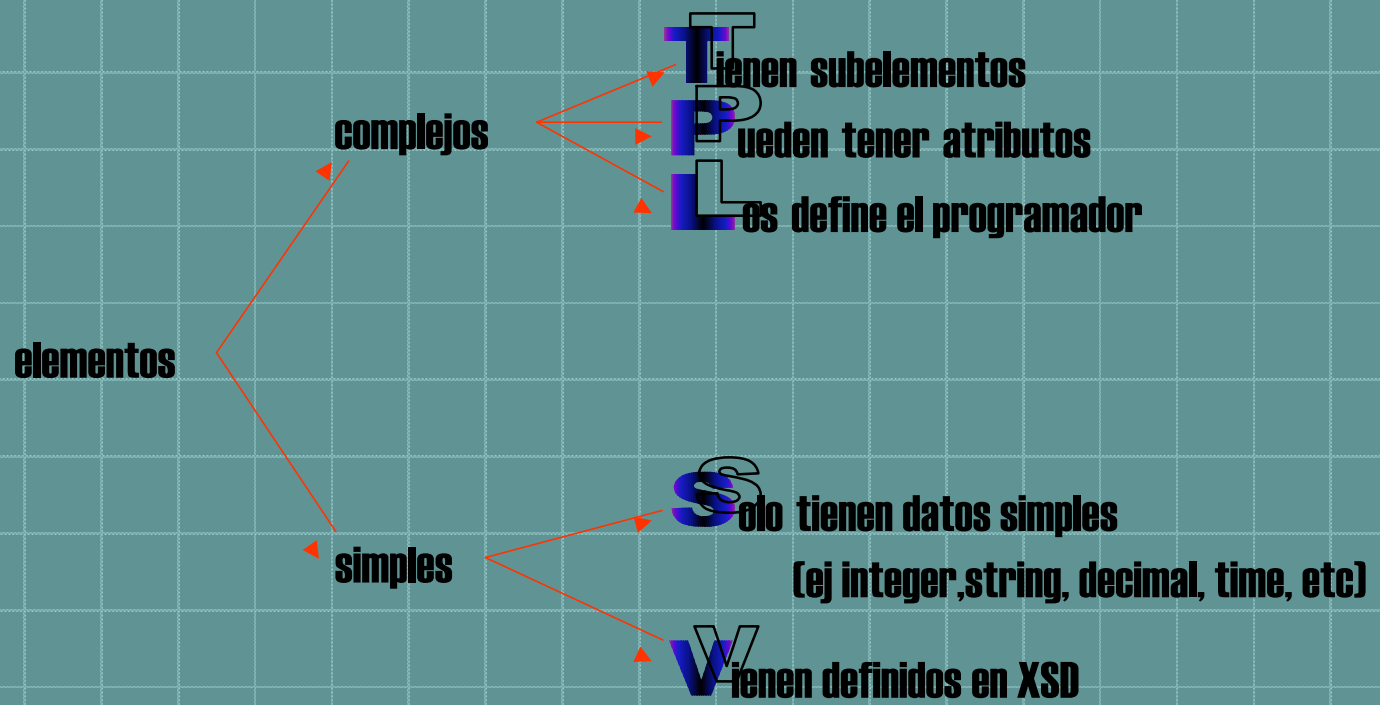
namespace

archivo de definición del schema

atributo perteneciente al namespace prefijado con xsi

namespace seguido con ":" para identificar el dominio del nombre

I.f. Tipos de elementos



1.g.Declaraciones de tipo

- Declaraciones de tipo para elementos simples:

```
<xsd:element name="price" type="xsd:string">
```

- Declaraciones de tipo para elementos comlejos:

```
<xsd:complexType name="price">
  <xsd:base="decimal"/>
  <xsd:attribute name="currency" type="string"
          minOccurs="1"  maxOccurs="unbounded" />
</xsd:complexType>
```

opcional

(sin límite)

default:

minOccurs= maxOccurs=1

EJEMPLO xsd VS dtd

(note.xml)

```
<?xml version="1.0"?>

<note >

  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

(note.xsd)

```
<note
xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.com note.xsd">
```

(note.dtd)

```
<!DOCTYPE note SYSTEM
"http://www.w3schools.com/dtd/note.dtd">
```

II. tecnologías XML

1. XSLT (transformations)
2. XPATH
3. XLINKS
4. XPointers
5. JAXP
6. SAX
7. DOM
8. XERCES

M. D. López De Luise

II.1. XSLT

eXtensible Style Language Transformation : (XSL Working Group/w3c)

lenguaje usado en XSL Style Sheets para transformar documentos fragmentos XML

A una distinta representación

EJEMPLO usando xslt

1 **Business-card.xml:**

```
<card type="simple">
  <name>John Doe</name>
   <title>CEO, Widget Inc.</title>
  <email>john.doe@widget.com</email>
  <phone>(202) 456-1414</phone>
</card>
```

John Doe
CEO, Widget Inc.
email: john.doe@widget.com
phone (202) 456 - 1414

EJEMPLO usando xslt

2 Definimos una semantica.XHTML para nuestra business-card usando una stylesheet XSLT :

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
xmlns="http://www.w3.org/1999/xhtml">
  <xsl:template match="card[@type='simple']">
    <html xmlns="http://www.w3.org/1999/xhtml">
      <title>business card</title><body>
        <xsl:apply-templates select="name" />
        <xsl:apply-templates select="title" />
        <xsl:apply-templates select="email" />
        <xsl:apply-templates select="phone" />
    </body></html>
  </xsl:template>

  <xsl:template match="card/name">
    <h1><xsl:value-of select="text()" /></h1>
  </xsl:template>
```

EJEMPLO usando xslt

```
<xsl:template match="email">
  <p>email: <a href="mailto:{text()}"><tt>
    <xsl:value-of select="text()"/>
  </tt></a></p>
  </xsl:template>
  ...
  </xsl:stylesheet>
```

3 **Resulta el documento: New-Business-card.xml**

```
<html xmlns="http://www.w3.org/1999/xhtml"><title>business card</title>
<body><h1>John Doe</h1><h3><i>CEO, Widget Inc.</i></h3>
<p>email: <a href="mailto:john.doe@widget.com"><tt>john.doe@widget.com</tt></a></p>
<p>phone: (202) 456-1414</p> </body></html>
```

4 **El browser mostrará:**

John Doe

CEO, Widget Inc.

email: john.doe@widget.com

phone: (202) 456-1414

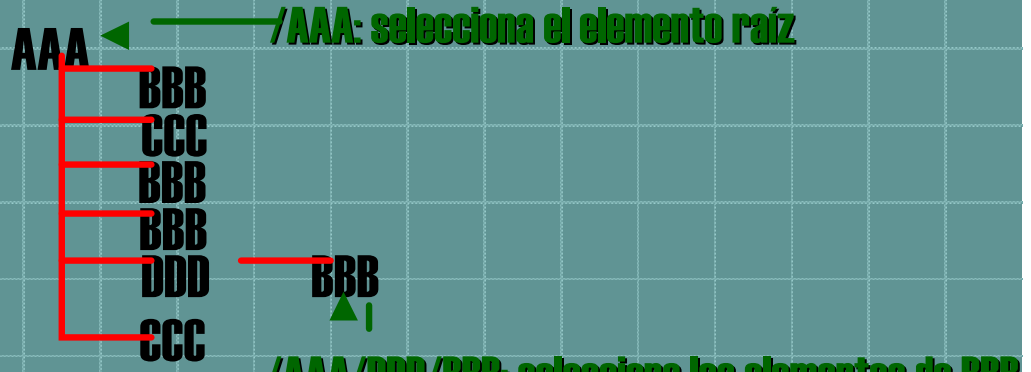
II.2. XPATH

Lenguaje para direccionar items en documentos xml

- usa la estructura lógica del documento
- simplifica el direccionamiento con XML
- entendible por XSLT y XPointer
- direcciona especificando la ruta del elemento
- direcciones locales con relaciones tipo ancestor, attribute, parent, etc.

EJEMPLO xpath

```
<AAA>
    <BBB/>
    <CCC/>
    <BBB/>
    <BBB/>
    <DDD>
        <BBB/>
    </DDD>
    <CCC/>
</AAA>
```



II.3. XLINKS (XML Linking Language)

Lenguaje que permite embeber links unidireccionales y bidireccionales hacia otros recursos en documentos XML

(ej. archivos, imágenes, documentos, programas, resultados de querys)

- XLink es un estándar W3C.
- XLink permite hacer links con archivos XML (o partes)
- XLink se basa en XPath.
- XLink permite implementar hiperlinks entre docs. XML (c/html)
- XLink permite asociar un link con metadatos

M. D. López De Luise

II.4. XPOINTERS

es una especificación que define una URI de un recurso específico de manera estándar.

Uniform Resource Identifiers

para recursos de tipo

- text/xml
- application/xml
- text/xml-external parsed entity
- application/xml-external-parsed-entity

M. D. López De Luise

EJEMPLO usando xpointer

java.lang.Object

java.net.URI

[scheme:]scheme-specific-part[#fragment]

1 URI opaca
- absoluta
- relativa

mailto:java-net@java.sun.com
news:comp.lang.java
urn:isbn:096139210x

2 URI jerárquica
- absoluta
- relativa

http://java.sun.com/j2se/1.3
/docs/guide/collections/designfaq.html#28
../../../demo/jfc/SwingSet2/src/SwingSet2.java
file:///~/calendar

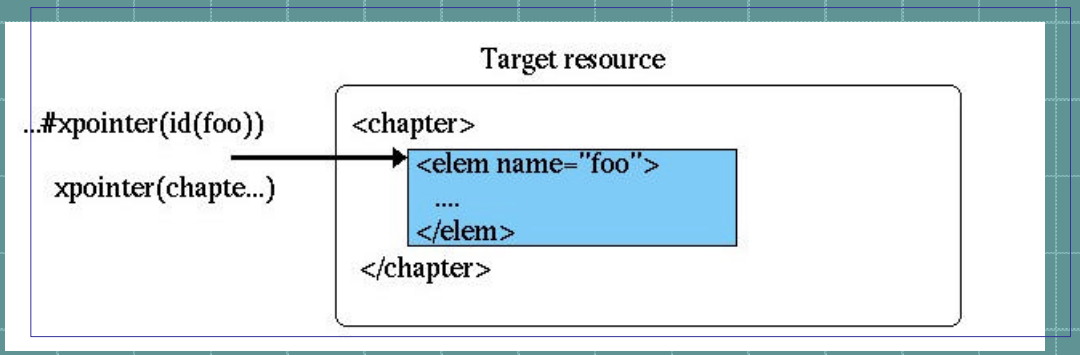
[user-info@]host[:port]

scheme-specific-part

[scheme:][//authority][path][?query][#fragment]

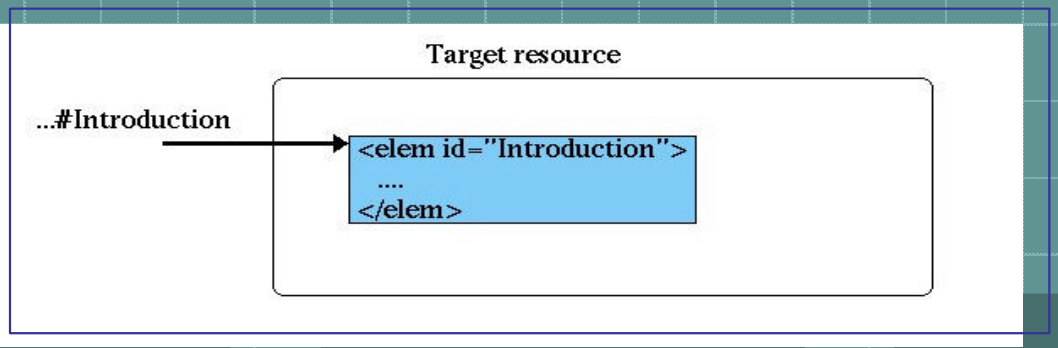
#xpointer(id("foo")) xpointer(/chapter[3]/elem[@name="foo"])

hace una búsqueda por ID



#Introduction

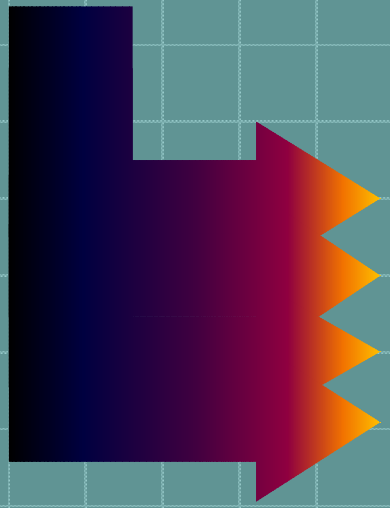
ubica un ID



II.5. JAXP (Java XML Pack)

http://java.sun.com/xml/jaxp/index.jsp

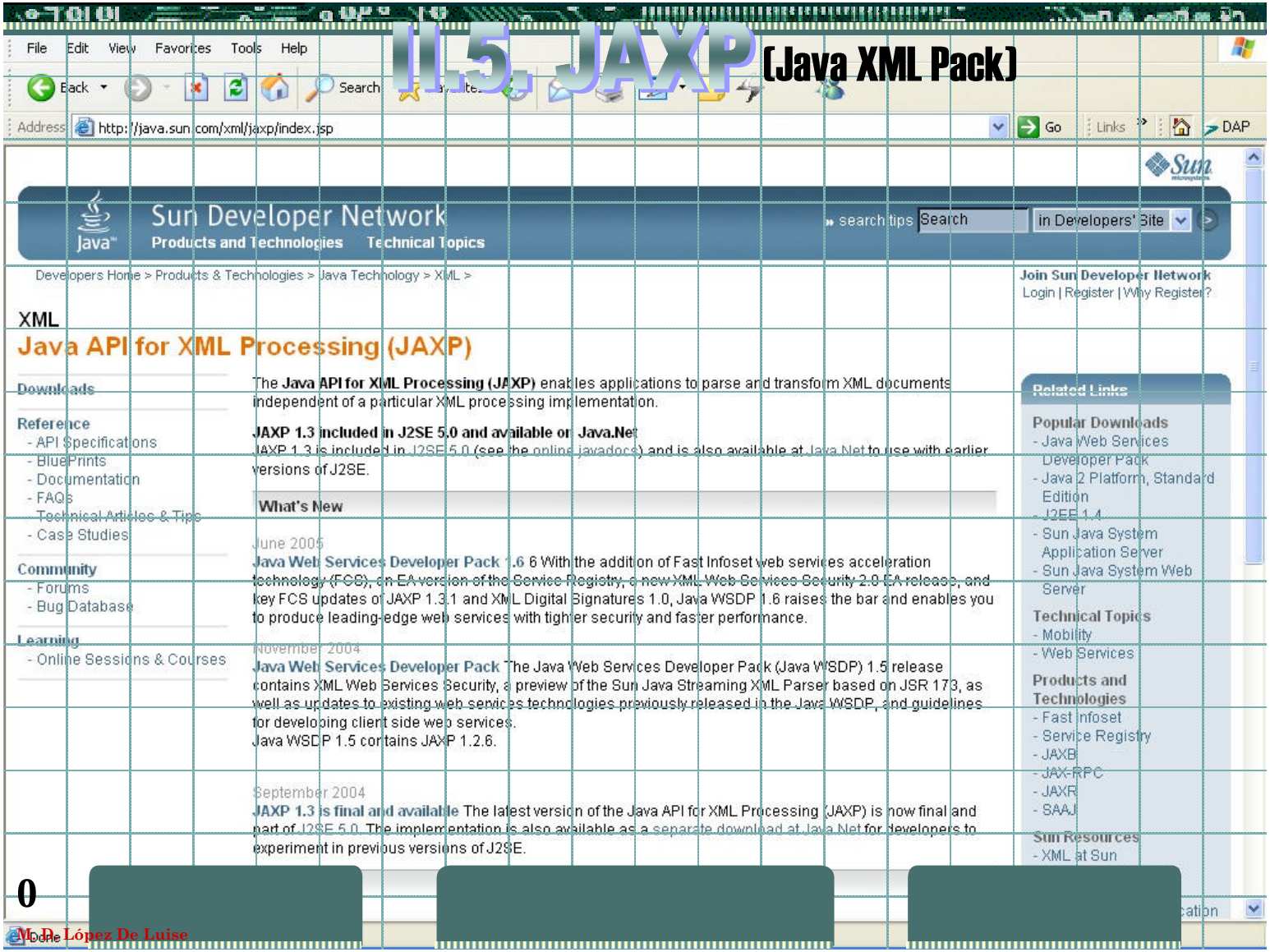
API for XML Processing (JAXP)



- interpreta XML (parsing)
- java API
- indep. de cualquier procesador XML
- manipula docs. XML con Java

M. D. López De Luise

II.5. JAXP (Java XML Pack)



M. De López De Luise

II.6. SAX y DOM

Formas de manejar XML programáticamente

Por eventos durante el parseo del documentno

esto hace

SAX

Parsear el documento y remodelizarlo en una estructura conocida por el programa

esto hace

DOM

II.6. SAX

- Verifica la validez de un XML
  - verifica que esté bien formado
    (tags sin terminar o mal escritos)
  - verifica que esté correcto contra un DTD o XML Schema
- Gratuito
- Permite manejar un XML programáticamente

II.6. SAX

def SAX: (Simple API for XML) permite insertar código de aplicación ante determinados eventos durante el parseo XML

ventaja: permite manejo secuencial del archivo sin necesidad de cargarlo íntegramente en memoria

para instalarlo:

bajarlo y copiar el sax.jar en JAVA_HOME/lib

II.6. SAX

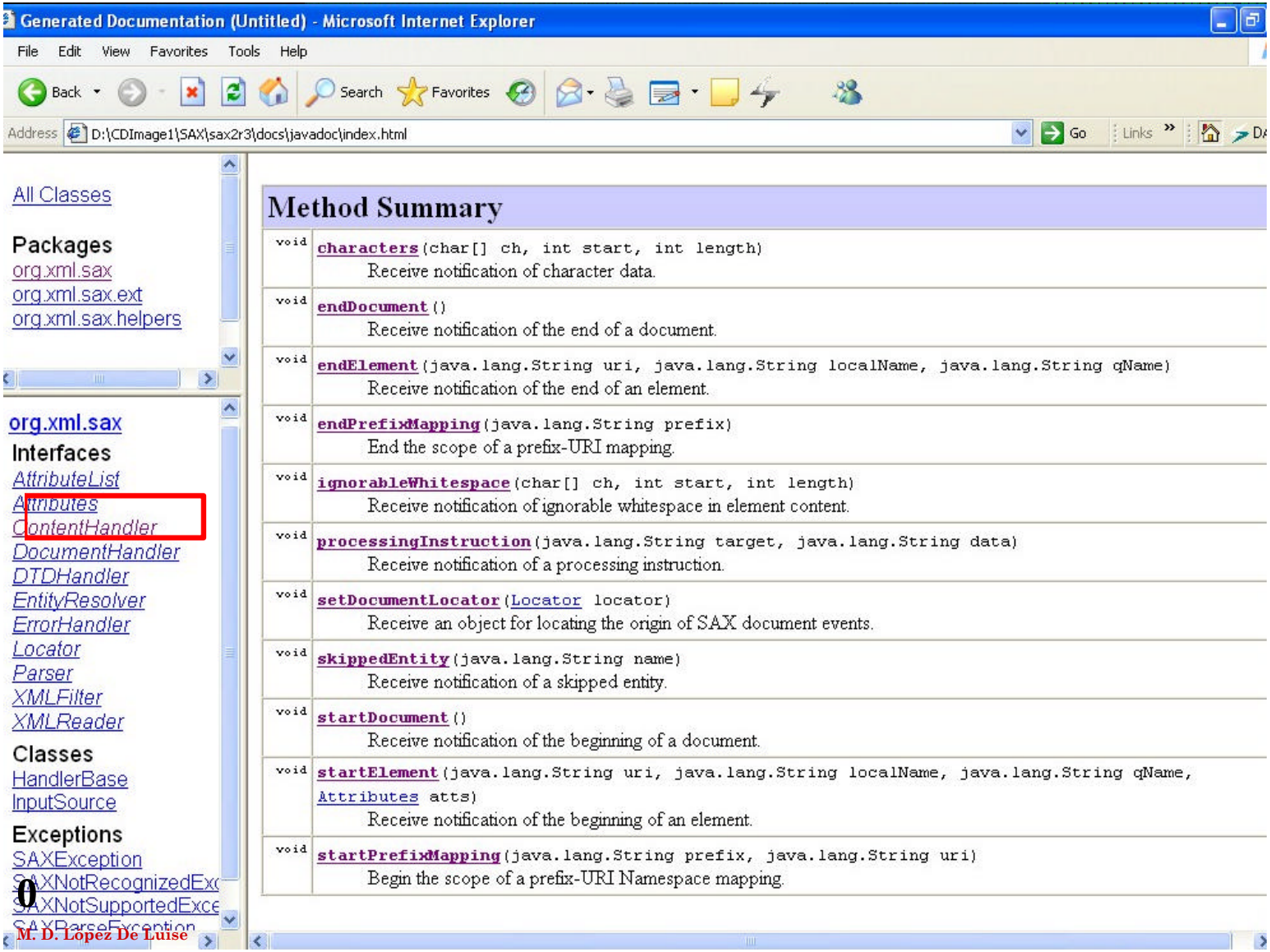
se manejan por métodos invocados por callback automáticamente por XMLparser durante el parsing

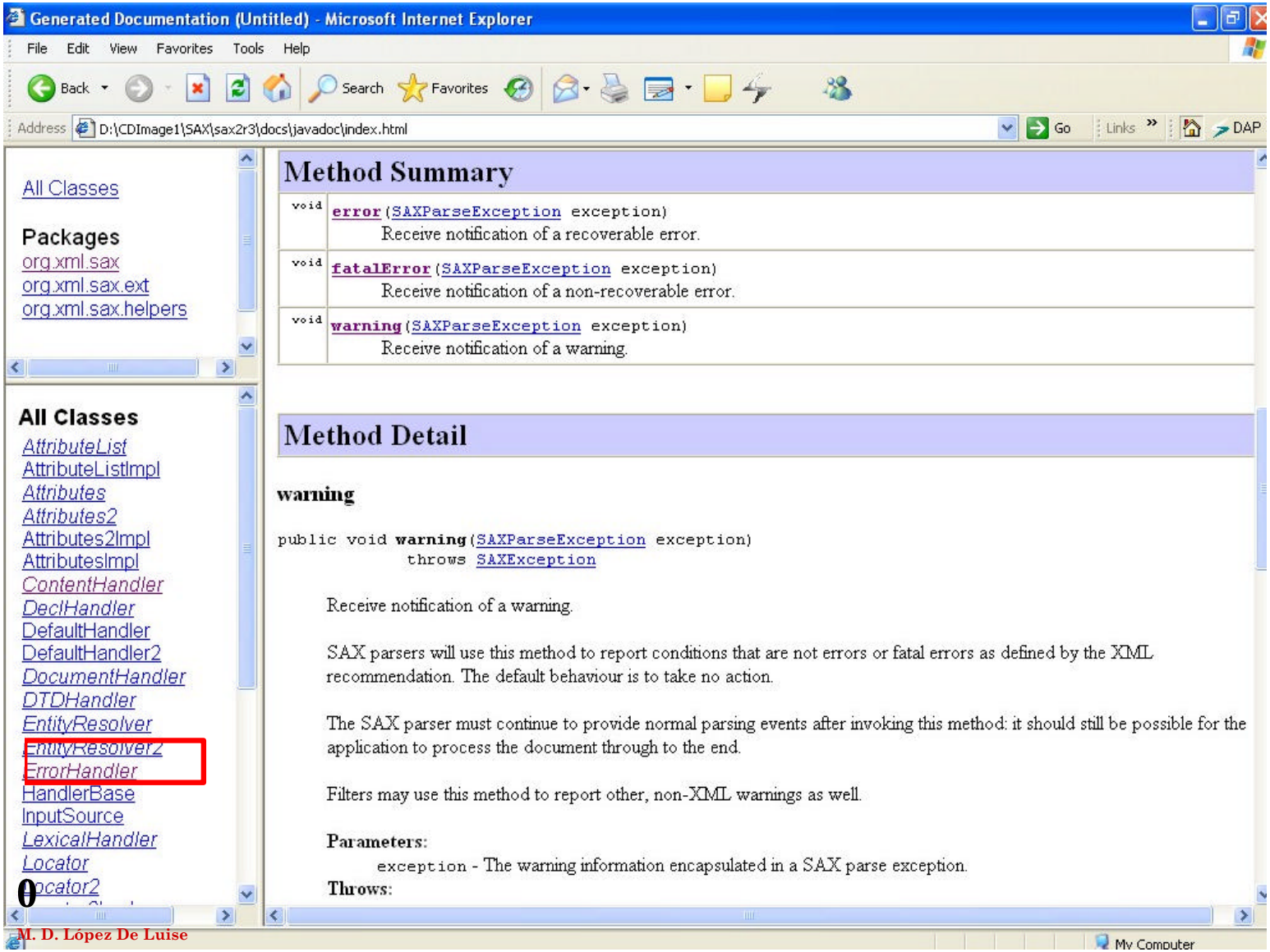
interfaces disponibles en SAX 2.0:

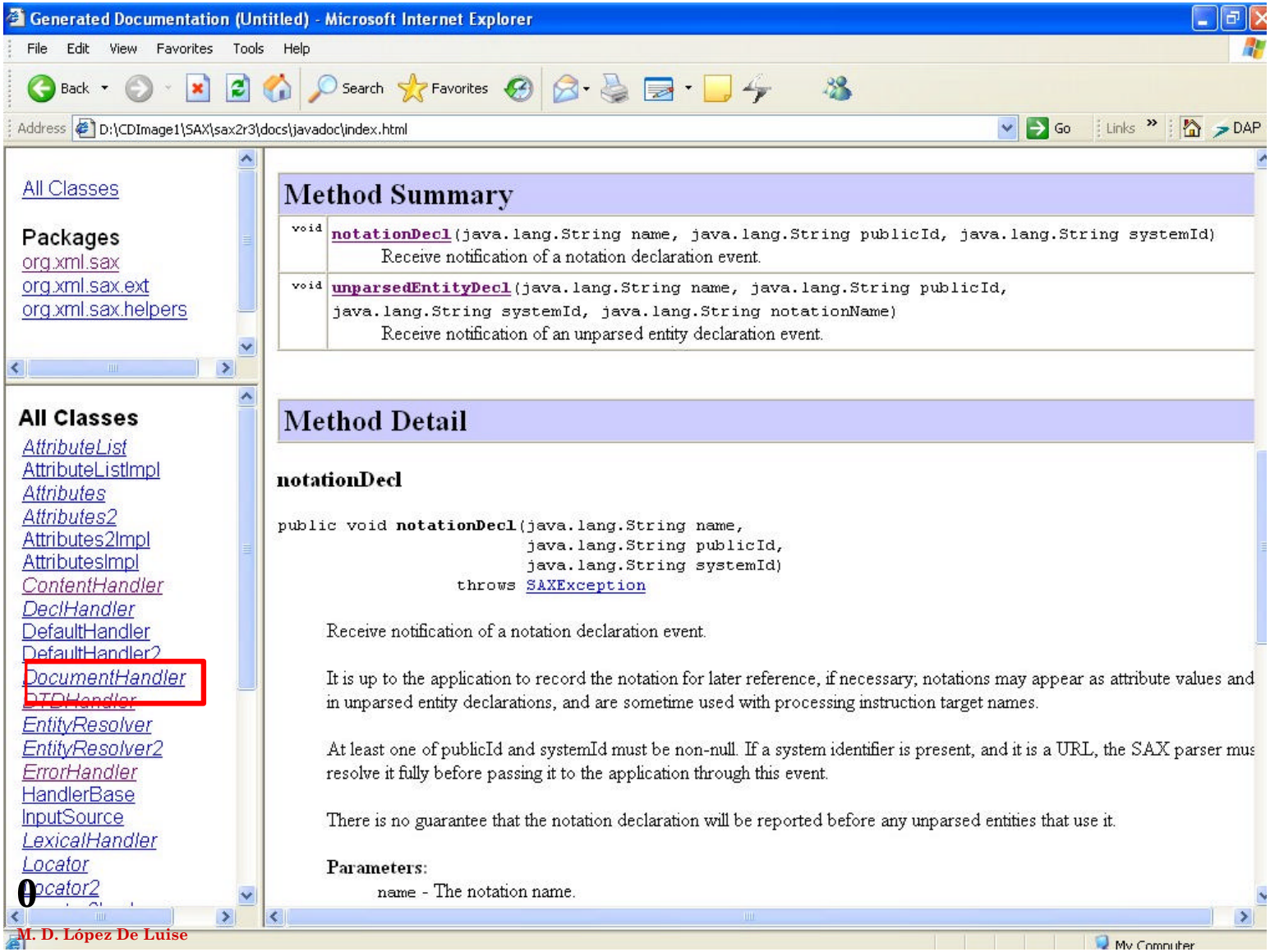
- org.xml.sax.ContentHandler
- org.xml.sax.ErrorHandler
- org.xml.sax.DTDHandler
- org.xml.sax.EntityResolver

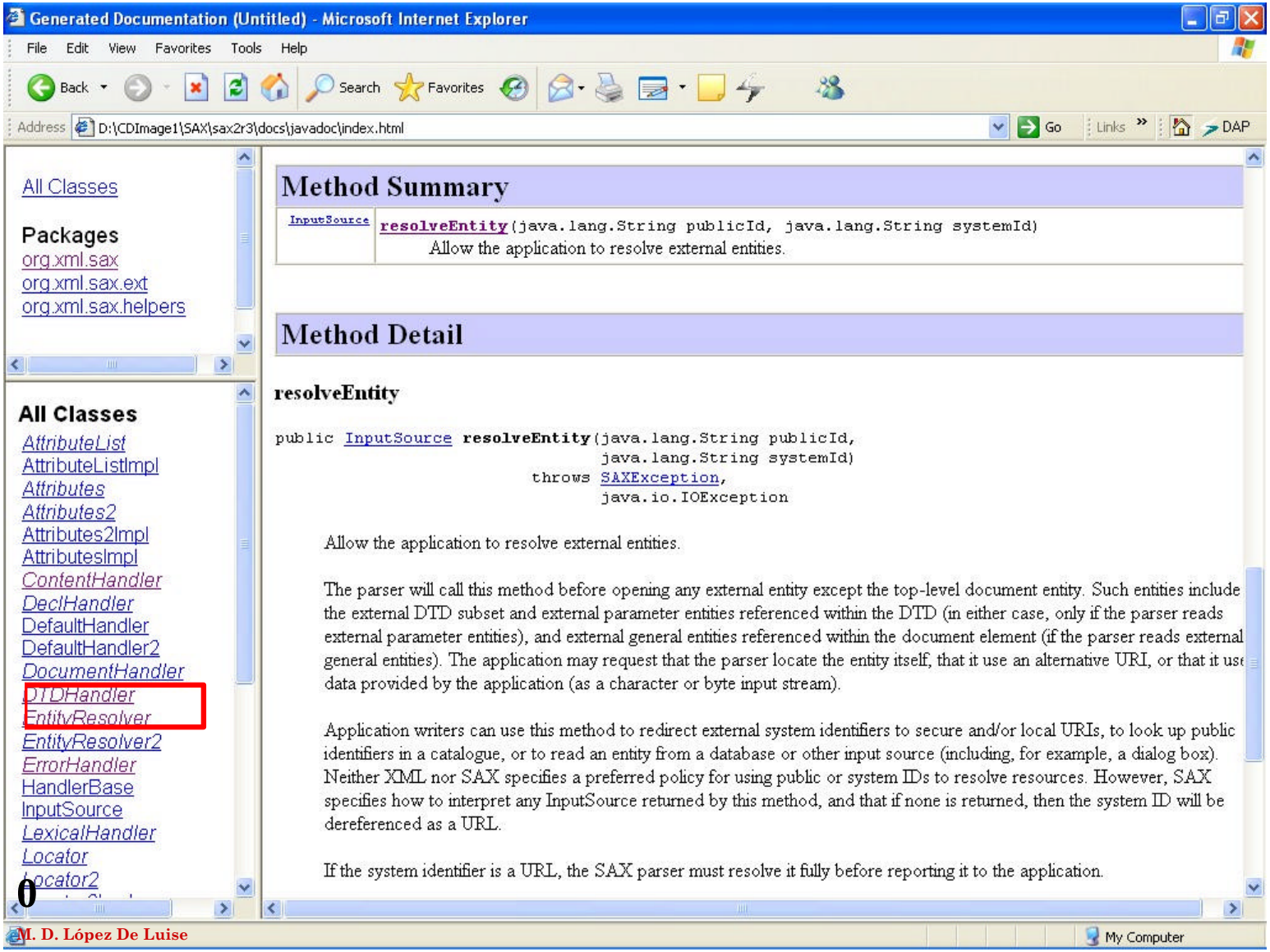
se registran con:

- parser.setContentHandler()
- parser.setErrorHandler()
- parser.setDTDHandler
- parser.setEntityResolver()


Generated Documentation (Untitled) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address D:\CDImage1\SAX\sax2r3\docs\javadoc\index.html
Go Links
All Classes
Packages
org.xml.sax
org.xml.sax.ext
org.xml.sax.helpers
org.xml.sax
Interfaces
AttributeList
Attributes
ContentHandler
DocumentHandler
DTDHandler
EntityResolver
ErrorHandler
Locator
Parser
XMLFilter
XMLReader
Classes
HandlerBase
InputSource
Exceptions
SAXException
SAXNotRecognizedExc
SAXNotSupportedExce
SAXParseException
Method Summary
void characters(char[] ch, int start, int length)
Receive notification of character data.
void endDocument()
Receive notification of the end of a document.
void endElement(java.lang.String uri, java.lang.String localName, java.lang.String qName)
Receive notification of the end of an element.
void endPrefixMapping(java.lang.String prefix)
End the scope of a prefix-URI mapping.
void ignorableWhitespace(char[] ch, int start, int length)
Receive notification of ignorable whitespace in element content.
void processingInstruction(java.lang.String target, java.lang.String data)
Receive notification of a processing instruction.
void setDocumentLocator(Locator locator)
Receive an object for locating the origin of SAX document events.
void skippedEntity(java.lang.String name)
Receive notification of a skipped entity.
void startDocument()
Receive notification of the beginning of a document.
void startElement(java.lang.String uri, java.lang.String localName, java.lang.String qName, Attributes atts)
Receive notification of the beginning of an element.
void startPrefixMapping(java.lang.String prefix, java.lang.String uri)
Begin the scope of a prefix-URI Namespace mapping.


Generated Documentation (Untitled) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address D:\CDImage1\SAX\sax2r3\docs\javadoc\index.html
Go Links DAP
All Classes
Packages
org.xml.sax
org.xml.sax.ext
org.xml.sax.helpers
All Classes
AttributeList
AttributeListImpl
Attributes
Attributes2
Attributes2Impl
AttributesImpl
ContentHandler
DeclHandler
DefaultHandler
DefaultHandler2
DocumentHandler
DTDHandler
EntityResolver
EntityResolver2
ErrorHandler
HandlerBase
InputSource
LexicalHandler
Locator
Locator2
Method Summary
void error(SAXParseException exception)
Receive notification of a recoverable error.
void fatalError(SAXParseException exception)
Receive notification of a non-recoverable error.
void warning(SAXParseException exception)
Receive notification of a warning.
Method Detail
warning
public void warning(SAXParseException exception)
throws SAXException
Receive notification of a warning.
SAX parsers will use this method to report conditions that are not errors or fatal errors as defined by the XML recommendation. The default behaviour is to take no action.
The SAX parser must continue to provide normal parsing events after invoking this method: it should still be possible for the application to process the document through to the end.
Filters may use this method to report other, non-XML warnings as well.
Parameters:
exception - The warning information encapsulated in a SAX parse exception.
Throws:
My Computer


Generated Documentation (Untitled) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address D:\CDImage1\SAX\sax2r3\docs\javadoc\index.html
Go Links DAP
All Classes
Packages
org.xml.sax
org.xml.sax.ext
org.xml.sax.helpers
All Classes
AttributeList
AttributeListImpl
Attributes
Attributes2
Attributes2Impl
AttributesImpl
ContentHandler
DeclHandler
DefaultHandler
DefaultHandler2
DocumentHandler
DTDHandler
EntityResolver
EntityResolver2
ErrorHandler
HandlerBase
InputSource
LexicalHandler
Locator
Locator2
Method Summary
void notationDecl(java.lang.String name, java.lang.String publicId, java.lang.String systemId)
Receive notification of a notation declaration event.
void unparsedEntityDecl(java.lang.String name, java.lang.String publicId, java.lang.String systemId, java.lang.String notationName)
Receive notification of an unparsed entity declaration event.
Method Detail
notationDecl
public void notationDecl(java.lang.String name, java.lang.String publicId, java.lang.String systemId) throws SAXException
Receive notification of a notation declaration event.
It is up to the application to record the notation for later reference, if necessary; notations may appear as attribute values and in unparsed entity declarations, and are sometime used with processing instruction target names.
At least one of publicId and systemId must be non-null. If a system identifier is present, and it is a URL, the SAX parser mus resolve it fully before passing it to the application through this event.
There is no guarantee that the notation declaration will be reported before any unparsed entities that use it.
Parameters:
name - The notation name.
My Computer


Generated Documentation (Untitled) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address D:\CDImage1\SAX\sax2r3\docs\javadoc\index.html
Go Links DAP
All Classes
Packages
org.xml.sax
org.xml.sax.ext
org.xml.sax.helpers
All Classes
AttributeList
AttributeListImpl
Attributes
Attributes2
Attributes2Impl
AttributesImpl
ContentHandler
DeclHandler
DefaultHandler
DefaultHandler2
DocumentHandler
DTDHandler
EntityResolver
EntityResolver2
ErrorHandler
HandlerBase
InputSource
LexicalHandler
Locator
Locator2
Method Summary
InputSource
resolveEntity(java.lang.String publicId, java.lang.String systemId)
Allow the application to resolve external entities.
Method Detail
resolveEntity
public InputSource resolveEntity(java.lang.String publicId,
java.lang.String systemId)
throws SAXException,
java.io.IOException
Allow the application to resolve external entities.
The parser will call this method before opening any external entity except the top-level document entity. Such entities include the external DTD subset and external parameter entities referenced within the DTD (in either case, only if the parser reads external parameter entities), and external general entities referenced within the document element (if the parser reads external general entities). The application may request that the parser locate the entity itself, that it use an alternative URI, or that it use data provided by the application (as a character or byte input stream).
Application writers can use this method to redirect external system identifiers to secure and/or local URIs, to look up public identifiers in a catalogue, or to read an entity from a database or other input source (including, for example, a dialog box). Neither XML nor SAX specifies a preferred policy for using public or system IDs to resolve resources. However, SAX specifies how to interpret any InputSource returned by this method, and that if none is returned, then the system ID will be dereferenced as a URL.
If the system identifier is a URL, the SAX parser must resolve it fully before reporting it to the application.
M. D. López De Luise
My Computer

II.7. DOM

validating parser

def DOM: Document Object Model, representación estándar del documento y el modela de documentos XML independientemente de las plataformas

características

- s API para XML documents
- ratuito
- omo todo parser permite manejar un XML programáticamente
- ay bindings en Java, JavaScript, C++, CORBA, Python, Perl, etc.
- onstruye en memoria un árbol de representación del documento completo
- os datos no son usables hasta que se terminó el parsing
- e supone que el arbol en memoria se procesará con una API DOM
- encilla manipulación de elementos del árbol (del documento modelado)

M. D. López De Luise

II.7. DOM

una breve historia

DOM level 0

- modelo de objetos no totalmente estándar de Explorer y Navigator

DOM level 1

- primer estándar para modelar objetos

DOM level 2

- organiza las interfaces
- agrega tratamiento de namespaces e interfaces p/eventos
- agrega rangos, views y style sheets
- es el estándar actualmente más distribuido

DOM level 3

- última versión
- agrega soporte para reconstruir un objeto Documento
- agrega soporte para procesar schemas DTD

M. D. López De Luise

II.7. DOM

Provee visibilidad OO del documento xml

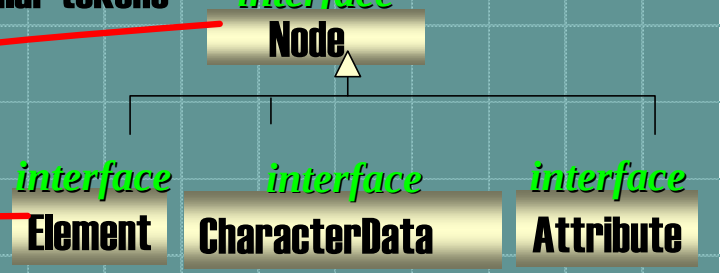
secuencia de caracteres → tokens → estructura en memoria

Provee api para recorrer la estructura y manipular tokens

ELEMENT_NODE
TEXT_NODE
CDATA_SECTION
PROCESSING_INSTRUCTION_NODE
...

Node parentNode()
Node[] childNodes()
...

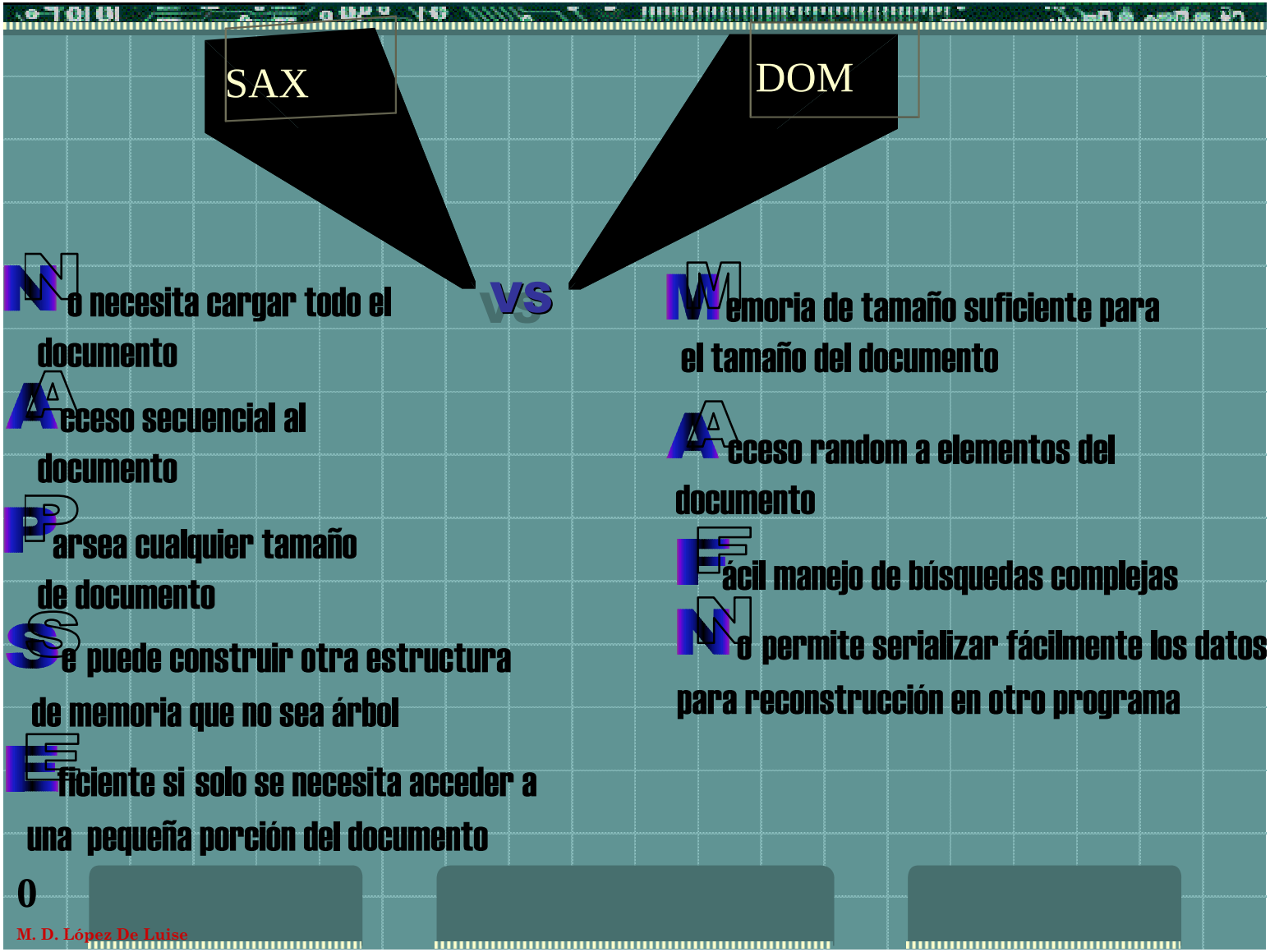
getDocumentElement()
getFirstChild()
getNextSibling()
...



métodos para administrar componentes de un Nodo
métodos para crear Nodo
métodos para modificar Nodo

ABM programático sobre el documento XML

M. D. López De Luise


SAX
DOM
VS
No necesita cargar todo el documento
Acceso secuencial al documento
Parsea cualquier tamaño de documento
Se puede construir otra estructura de memoria que no sea árbol
Eficiente si solo se necesita acceder a una pequeña porción del documento
Memoria de tamaño suficiente para el tamaño del documento
Acceso random a elementos del documento
Fácil manejo de búsquedas complejas
No permite serializar fácilmente los datos para reconstrucción en otro programa

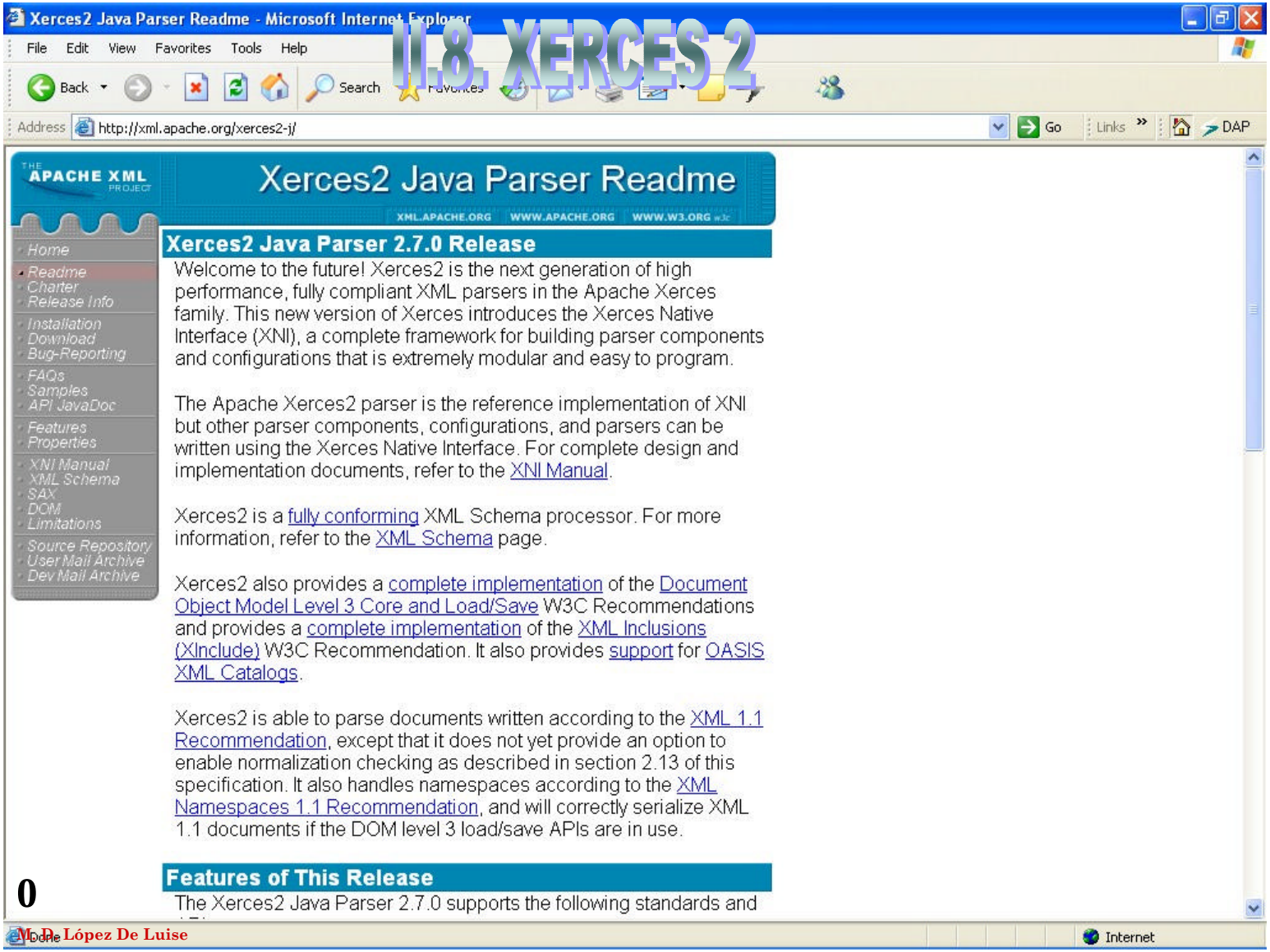
II.3. XERCES 2

validating parser

- Trabaja con SAX 2.0 + DOM Level 2
- Como parser Java se lo denomnina xerces-J
- Gratuito: xml.apache.org
- Como todo parser permite manejar un XML programáticamente
- Permite serializar

M. D. López De Luise

11.8. XERCES 2


Xerces2 Java Parser Readme - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://xml.apache.org/xerces2-j/
Go Links DAP
THE APACHE XML PROJECT
Xerces2 Java Parser Readme
XML.APACHE.ORG WWW.APACHE.ORG WWW.W3.ORG
Home
Readme
Charter
Release Info
Installation
Download
Bug-Reporting
FAQs
Samples
API JavaDoc
Features
Properties
XNI Manual
XML Schema
SAX
DOM
Limitations
Source Repository
User Mail Archive
Dev Mail Archive
Xerces2 Java Parser 2.7.0 Release
Welcome to the future! Xerces2 is the next generation of high performance, fully compliant XML parsers in the Apache Xerces family. This new version of Xerces introduces the Xerces Native Interface (XNI), a complete framework for building parser components and configurations that is extremely modular and easy to program.
The Apache Xerces2 parser is the reference implementation of XNI but other parser components, configurations, and parsers can be written using the Xerces Native Interface. For complete design and implementation documents, refer to the XNI Manual.
Xerces2 is a fully conforming XML Schema processor. For more information, refer to the XML Schema page.
Xerces2 also provides a complete implementation of the Document Object Model Level 3 Core and Load/Save W3C Recommendations and provides a complete implementation of the XML Inclusions (XInclude) W3C Recommendation. It also provides support for OASIS XML Catalogs.
Xerces2 is able to parse documents written according to the XML 1.1 Recommendation, except that it does not yet provide an option to enable normalization checking as described in section 2.13 of this specification. It also handles namespaces according to the XML Namespaces 1.1 Recommendation, and will correctly serialize XML 1.1 documents if the DOM level 3 load/save APIs are in use.
Features of This Release
The Xerces2 Java Parser 2.7.0 supports the following standards and
Done
Internet

III. Serialización

serializar → pasarlo por una network/text file/Output Stream

con xerces-J viene el package serialize que serializa (basado sólo en DOM docums DOM sobre OutputStreams.

DOMBuilder con las interfaces DOMWriter

con DOM 3: incluye package para Load y Save para administrar documetnos DOM a través de programas de manera independientede la implementacion

ventaja contra XMLSerializer: no se limita a volcar documentos, fragmentos o elementos. Puede volcar a cualquier tipo de nodo y con filtros personlaizados

IV. Protocolos XML

1. XML RPC
2. JAX-RPC
3. SOAP

M. D. López De Luise

IV. Protocolos XML

XML-RPC

DEF: es un protocolo que implementa el RPC de Java codificado en XML.

Tipos de datos:

vector sin claves

```
<array><data>
   <value><i4>14</i4></value>
   <value><string>esta es una pba</string></value>
   <value><i4>1900</i4></value>
</data> </array>
```

dato binario en base 64

```
<base64>eW91IGNhbid0IHJlYWQgdGhpcyE=</base64>
```

dato booleano (0 o 1)

```
<boolean>1</boolean>
```

dato date/time

```
<dateTime.iso8601>19980717T14:08:55</dateTime.iso8601>
```

dato precisión double

```
<double>-12.53</double>
```

dato integer

```
<i4>42</i4>
```

dato String

```
<string>Hello world!</string>
```

vector con claves

```
<struct>
<member><name>d1</name> <value><i4>1</i4></value></member>
<member><name>d2</name><value><i4>2</i4></value></member>
</struct>
```

IV. Protocolos XML

XML-RPC: JAX-RPC

DEF: (Java API for XML-based RPC) es un service endpoint definido e implementado en Java.

usable por clientes en cualquier lenguaje/ plataforma

clientes Java pueden invocar RPC - XML de plataformas no Java

EJEMPLO Service endpoint JAX-RPC:

Service Definition Interface: StockQuoteProvider

```
package com.example;
public interface StockQuoteProvider extends java.rmi.Remote {
   public float getLastTradePrice (String tickerSymbol) throws java.rmi.RemoteException;
   // .. otros metodos remotos
}
```

IV. Protocolos XML

XML-SOAP

def: Simple Object Access ProtocoD, protocolo liviano de mensajería basado en XML.

características:

- usado para Web services
- independiente del S.O.
- compatible con una serie de protocolos
  - SMTP
  - MIME
  - HTTP

M. D. López De Luise

Aplicaciones

ServiceGlobe

UDDI* para administrar servicios

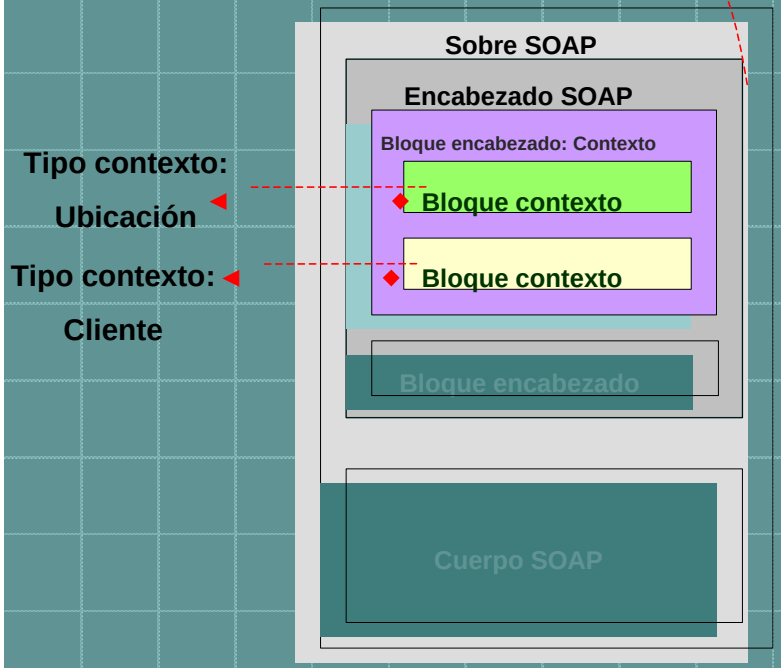
*UDDI: Universal, Description, Discovery and Integration

Aplicaciones

ServiceGlobe

Mensaje SOAP* para solicitar servicios al **HS**

El modelo de contexto

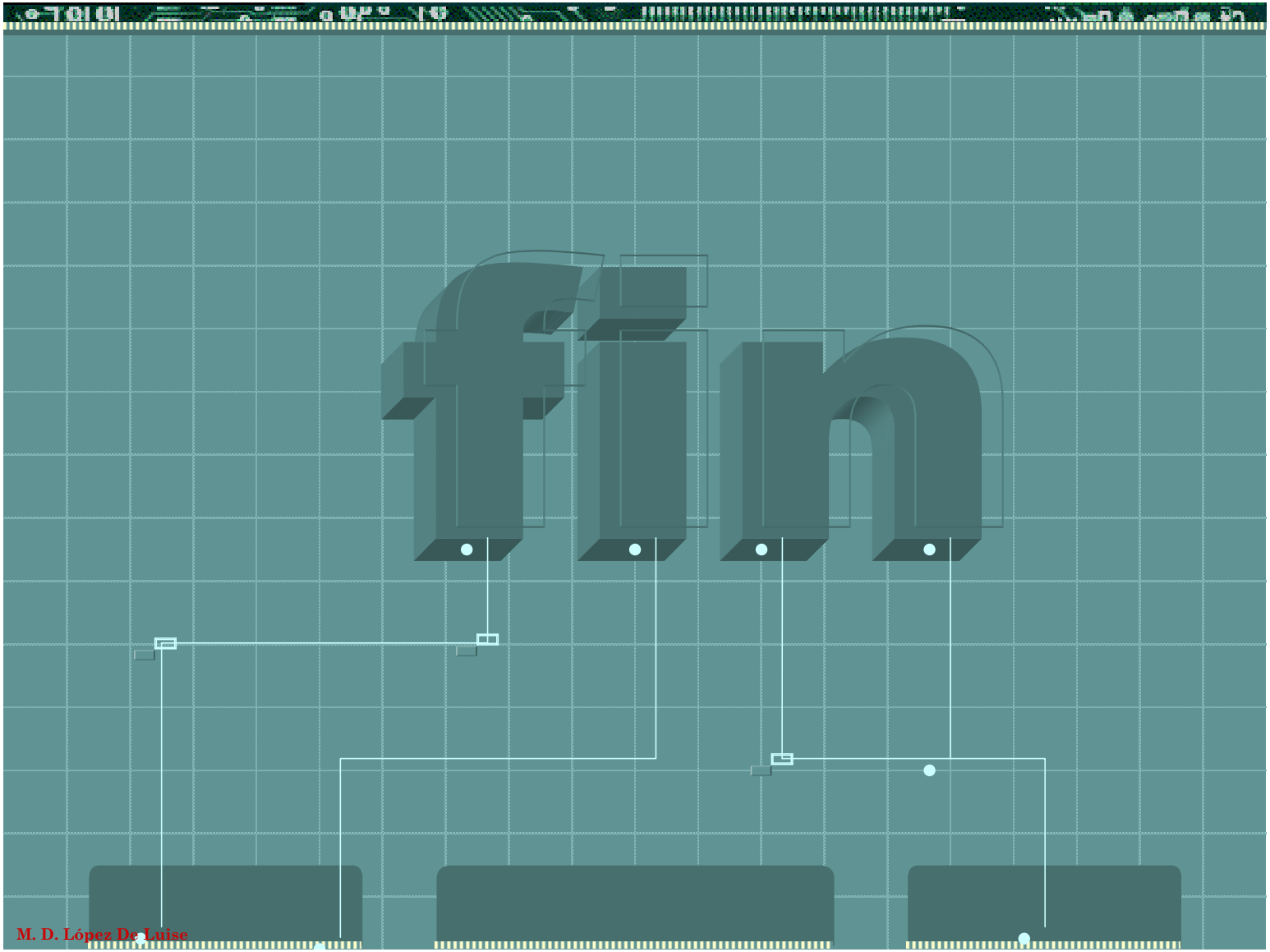


Tipo contexto:
Ubicación

Tipo contexto:
Cliente

```
<env:Envelope xmlns:env="..." >
 <env:Header>
  <Context
    xmlns="http://sg.fmi.uni-passau.de/context">
   <Client>
    <Hardware>
     <Defaults>
      http://example.com/PDA
     </Defaults>
     <ScreenSize>320x320</ScreenSize>
     <IsColorCapable>Yes</IsColorCapable>
    </Hardware>
   </Client>
  </Context>
 </env:Header>
 <env:Body>
  <!-- serialized object data -->
 </env:Body>
</env:Envelope>
```


SOAP:Simple Object Access Protocol


fin
M. D. López De Luise