Reducción de la DTD DocBook XML v4.1.2

Fernando Reyero Noya

fernando.reyero@hispalinux.es

Sergio González González

sergio.gonzalez@hispalinux.es

Documentación que detalla el proceso de reducción de la DTD DocBook XML v4.1.2 a las etiquetas más comúnmente utilizadas.

Introducción

La finalidad de este proyecto es la reducción de la DTD DocBook XML v4.1.2 (http://www.oasis-open.org/docbook/xml/4.1.2/index.shtml) a aquellas etiquetas más comúnmente utilizadas. Para ello analizamos una serie de documentos escritos por distintos autores[1], para hacer la muestra lo más heterogénea posible.

Los documentos utilizados se encuentran en el directorio documentos (../documentos/documentos.html), repartidos entre dos subdirectorios:

- docbook_xml (../documentos/docbook_xml/), bajo el cual están almacenados los documentos que utilizan la DTD "a reducir".

  Nota: Estos son los documentos que se han utilizado para obtener las etiquetas (a partir de ahora nos referiremos a ellos como la muestra utilizada)

- docbook_xml_reducida (../documentos/docbook_xml_reducida/), ubicación en la que se encuentran los mismos documentos que en el anterior subdirectorio, salvo por la DTD utilizada. Estos documentos tienen como DTD la obtenida tras la realización de este proyecto, es decir, la DTD Reducida[2].

Como se ha realizado

El proyecto se ha dividido en dos partes:

1. Obtención de etiquetas y el número de repeticiones de cada etiqueta, a partir de la muestra considerada.
2. Reducción de la DTD DocBook XML v4.1.2. La nueva DTD contendrá únicamente aquellas etiquetas obtenidas en la primera parte.

Obtención de las etiquetas

La búsqueda de las etiquetas dentro de un documento, se llevó a cabo con el programa analizador_tags (../codigo/codigo.html)[3], y el número de veces que aparece una determinada etiqueta en la muestra se obtiene gracias al script contador_tags.pl (../codigo/codigo.html)[4].

El script principal del proyecto es el denominado “contador_tags.pl”. Este se encarga de buscar los documentos para analizar, verificar que están correctamente formados[5], obtener las etiquetas que estos posean[6] y mostrar por pantalla la relación de etiquetas más comúnmente utilizadas, junto al número de repeticiones de cada.

El resultado obtenido, tras ejecutar el script “contador_tags.pl” sobre el directorio /documentos/docbook_xml/ (../documentos/docbook_xml/)[7], es:

```
[fys@todoscsi]$ ./contador_tags.pl


Buscando los documentos... [Hecho]

Analizando la validez de los documentos... [Hecho]

Buscando etiquetas... [Hecho]

Contando las etiquetas... [Hecho]

A continuación se mostrarán las etiquetas y el número de apariciones:

Número      Etiquetas
apariciones

1     constant
1     constructorsynopsis
1     fax
1     fieldsynopsis
1     glossary
1     glossaryinfo
1     glossseealso
1     informalfigure
1     initializer
1     markup
1     menuchoice
1     modespec
1     otheraddr
```

1	phone
1	reference
1	referenceinfo
1	sectioninfo
1	set
1	setinfo
1	tfoot
1	toclevel2
1	tocpart
2	ackno
2	artpagenums
2	bibliodiv
2	bibliographyinfo
2	bibliomixed
2	classname
2	classsynopsis
2	collab
2	collabname
2	confdates
2	confgroup
2	confnum
2	confsponsor
2	conftitle
2	contractnum
2	contractsponsor
2	corpauthor
2	epigraph
2	errorname
2	honorific
2	invpartnumber
2	lineage
2	olink
2	ooclass
2	printhistory
2	pubsnumber
2	refsynopsisdiv
2	seriesvolnums
2	void
3	city
3	country
3	equation
3	glossdiv
3	guibutton
3	index
3	keysym
3	keywordset
3	postcode
3	seealsoie
3	segmentedlist
3	simpara
3	state
3	street
3	synopsis

3	`titleabbrev`
3	`tocchap`
3	`toclevel1`
4	`bibliomset`
4	`biblioset`
4	`firstterm`
4	`inlinegraphic`
4	`issn`
4	`issuenum`
4	`methodsynopsis`
4	`pagenums`
4	`qandadiv`
4	`shortaffil`
4	`volumenum`
5	`authorblurb`
5	`bibliography`
5	`dedication`
5	`isbn`
5	`jobtitle`
5	`methodname`
5	`refmeta`
5	`returnvalue`
6	`areaset`
6	`guisubmenu`
6	`methodparam`
6	`orgdiv`
6	`toc`
7	`areaspec`
7	`edition`
7	`informalexample`
7	`othername`
7	`programlistingco`
7	`refentry`
7	`refnamediv`
7	`refpurpose`
7	`refsect1`
7	`seglistitem`
7	`symbol`
8	`anchor`
8	`attribution`
8	`bridgehead`
8	`caption`
8	`group`
8	`publisher`
8	`sect5`
8	`sgmltag`
8	`superscript`
9	`legalnotice`
10	`corpname`
10	`editor`
10	`guimenu`
10	`keycombo`
10	`preface`

10	refname
10	subtitle
11	cmdsynopsis
11	partintro
11	varname
12	foreignphrase
12	modifier
12	publishername
12	qandaset
13	segtitle
14	productnumber
14	trademark
15	type
17	caution
17	optional
17	part
18	keyword
19	copyright
19	holder
20	arg
20	authorgroup
20	citetitle
21	hardware
21	see
21	seg
22	bookinfo
22	graphic
22	warning
23	chapterinfo
24	orgname
24	tocentry
26	book
26	citation
27	biblioentry
27	year
28	pubdate
28	token
30	revhistory
32	appendix
32	sidebar
33	othercredit
34	contrib
34	highlights
34	substeps
35	area
35	glosslist
37	keycap
38	parameter
39	informaltable
39	table
43	abstract
44	thead
47	qandaentry

```
47      question
47      wordasword
48      answer
49      blockquote
50      seeie
51      calloutlist
52      productname
56      important
56      structfield
66      figure
71      guimenuitem
74      function
75      tertiaryie
78      tbody
78      tgroup
79      tertiary
81      tip
87      abbrev
95      colspec
104     procedure
105     example
122     articleinfo
126     authorinitials
128     article
130     revremark
132     callout
132     releaseinfo
134     revision
134     revnumber
138     note
147     mediaobject
150     co
151     date
161     simplelist
162     chapter
166     orderedlist
174     affiliation
174     citerefentry
175     prompt
177     formalpara
179     manvolnum
179     refentrytitle
185     address
188     application
196     author
199     variablelist
215     email
224     footnote
244     surname
248     firstname
260     option
264     inlinemediaobject
273     computeroutput
```

```
311     quote
384     textobject
390     link
393     xref
403     phrase
410     itemizedlist
435     literallayout
443     imagedata
443     imageobject
453     step
461     glossdef
461     glossentry
465     glossterm
480     row
511     sect3
531     replaceable
549     section
551     envar
658     programlisting
742     sect1
862     userinput
922     varlistentry
935     term
998     indexentry
998     primaryie
1035    secondaryie
1084    sect2
1200    secondary
1259    sect4
1273    entry
1358    systemitem
1511    emphasis
1591    acronym
1799    screen
1951    literal
2318    primary
2429    indexterm
2756    member
3328    ulink
3453    listitem
4198    command
5279    title
6077    filename
14982   para
[fys@todoscsi]$
```

Sugerencia: Esta salida no es exactamente la información que presenta el script. “contador_tags.pl” no ordena las etiquetas de menor a mayor frecuencia de aparición, simplemente muestra las etiquetas según las va procesando.

La ordenación se ha obtenido gracias al programa sort, para mejorar la legibilidad de la salida.

Reducción de la DTD

Una vez obtenidas las etiquetas que permanecerían en la DTD, se obtuvo un listado de las etiquetas que debían ser eliminadas de la DTD DocBook XML v4.1.2, debido a su poco uso.

Estas etiquetas pueden verse en el archivo `rinclx.mod` de la DTD Reducida (../dtd/dtd.html). El contenido del archivo anterior se muestra a continuación:

```
<!--

  ######################################################################

  DocBook XML Reducida Incorporacion V4.1.2.1
  Este archivo es una parte de DocBook XML DTD Reducida V4.1.2.1

  Cualquier duda o comentario sobre esta DTD dirijalas a:

    Fernando Reyero <fernando.reyero@hispalinux.es>
    Sergio Gonzalez <sergio.gonzalez@hispalinux.es>

  ######################################################################

-->


<!-- Entidades de la piscina de de informacion  -->

<!ENTITY % bibliomisc.module "IGNORE">
<!ENTITY % subjectset.content.module "IGNORE">
<!ENTITY % itermset.module "IGNORE">
<!ENTITY % msgset.content.module "IGNORE">
<!ENTITY % msgentry.module "IGNORE">
<!ENTITY % simplemsgentry.module "IGNORE">
<!ENTITY % msg.module "IGNORE">
<!ENTITY % msgmain.module "IGNORE">
<!ENTITY % msgsub.module "IGNORE">
<!ENTITY % msgrel.module "IGNORE">
<!ENTITY % msginfo.module "IGNORE">
<!ENTITY % msglevel.module "IGNORE">
<!ENTITY % msgorig.module "IGNORE">
<!ENTITY % msgaud.module "IGNORE">
<!ENTITY % msgexplan.module "IGNORE">
<!ENTITY % label.module "IGNORE">
<!ENTITY % sidebarinfo.module "IGNORE">
<!ENTITY % remark.module "IGNORE">
<!ENTITY % glosssee.module "IGNORE">
<!ENTITY % screenco.module "IGNORE">
<!ENTITY % screenshot.content.module "IGNORE">
<!ENTITY % screeninfo.module "IGNORE">
<!ENTITY % graphicco.module "IGNORE">
<!ENTITY % videoobject.module "IGNORE">
<!ENTITY % audioobject.module "IGNORE">
<!ENTITY % objectinfo.module "IGNORE">
<!ENTITY % videodata.module "IGNORE">
```

```
<!ENTITY % audiodata.module "IGNORE">
<!ENTITY % mediaobjectco.module "IGNORE">
<!ENTITY % imageobjectco.module "IGNORE">
<!ENTITY % informalequation.module "IGNORE">
<!ENTITY % inlineequation.module "IGNORE">
<!ENTITY % alt.module "IGNORE">
<!ENTITY % sbr.module "IGNORE">
<!ENTITY % synopfragmentref.module "IGNORE">
<!ENTITY % synopfragment.module "IGNORE">
<!ENTITY % funcsynopsis.content.module "IGNORE">
<!ENTITY % funcsynopsisinfo.module "IGNORE">
<!ENTITY % funcprototype.module "IGNORE">
<!ENTITY % funcdef.module "IGNORE">
<!ENTITY % varargs.module "IGNORE">
<!ENTITY % paramdef.module "IGNORE">
<!ENTITY % funcparams.module "IGNORE">
<!ENTITY % classsynopsisinfo.module "IGNORE">
<!ENTITY % oointerface.module "IGNORE">
<!ENTITY % ooexception.module "IGNORE">
<!ENTITY % interfacename.module "IGNORE">
<!ENTITY % exceptionname.module "IGNORE">
<!ENTITY % destructorsynopsis.module "IGNORE">
<!ENTITY % pob.module "IGNORE">
<!ENTITY % revdescription.module "IGNORE">
<!ENTITY % accel.module "IGNORE">
<!ENTITY % action.module "IGNORE">
<!ENTITY % database.module "IGNORE">
<!ENTITY % errorcode.module "IGNORE">
<!ENTITY % errortype.module "IGNORE">
<!ENTITY % guiicon.module "IGNORE">
<!ENTITY % guilabel.module "IGNORE">
<!ENTITY % interface.module "IGNORE">
<!ENTITY % keycode.module "IGNORE">
<!ENTITY % lineannotation.module "IGNORE">
<!ENTITY % medialabel.module "IGNORE">
<!ENTITY % shortcut.module "IGNORE">
<!ENTITY % mousebutton.module "IGNORE">
<!ENTITY % msgtext.module "IGNORE">
<!ENTITY % property.module "IGNORE">
<!ENTITY % structname.module "IGNORE">
<!ENTITY % footnoteref.module "IGNORE">
<!ENTITY % beginpage.module "IGNORE">


<!-- Entidades de la organizacion de la documentacion  -->


<!ENTITY % local.indexdivcomponent.mix "IGNORE">
<!ENTITY % colophon.module "IGNORE">
<!ENTITY % tocfront.module "IGNORE">
<!ENTITY % toclevel3.module "IGNORE">
<!ENTITY % toclevel4.module "IGNORE">
<!ENTITY % toclevel5.module "IGNORE">
```

```
<!ENTITY % tocback.module "IGNORE">
<!ENTITY % lot.content.module "IGNORE">
<!ENTITY % lotentry.module "IGNORE">
<!ENTITY % appendixinfo.module "IGNORE">
<!ENTITY % indexinfo.module "IGNORE">
<!ENTITY % setindexinfo.module "IGNORE">
<!ENTITY % partinfo.module "IGNORE">
<!ENTITY % prefaceinfo.module "IGNORE">
<!ENTITY % refentryinfo.module "IGNORE">
<!ENTITY % refsect1info.module "IGNORE">
<!ENTITY % refsect2info.module "IGNORE">
<!ENTITY % refsect3info.module "IGNORE">
<!ENTITY % refsynopsisdivinfo.module "IGNORE">
<!ENTITY % local.sect1info.attrib "IGNORE">
<!ENTITY % local.sect2info.attrib "IGNORE">
<!ENTITY % local.sect3info.attrib "IGNORE">
<!ENTITY % local.sect4info.attrib "IGNORE">
<!ENTITY % local.sect5info.attrib "IGNORE">
<!ENTITY % simplesect.module "IGNORE">
<!ENTITY % indexes.module "IGNORE">
<!ENTITY % indexdiv.module "IGNORE">
<!ENTITY % refmiscinfo.module "IGNORE">
<!ENTITY % refdescriptor.module "IGNORE">
<!ENTITY % refclass.module "IGNORE">
<!ENTITY % refsect2.module "IGNORE">
<!ENTITY % refsect3.module "IGNORE">


<!--

  ######################################################################

          Fin de DocBook XML Reducida Incorporacion V4.1.2.1

  ######################################################################

-->
```

Nota: La eliminación de una determinada etiqueta de la DTD “original”, implica comprobar que desaparece todo rastro de dicha etiqueta en la DTD y que esta sigue manteniéndose consistente.

Detallando los archivos de la DTD DocBook XML Reducida v4.1.2.1

Tras eliminar todas las etiquetas que no se usan de la DTD DocBook XML v4.1.2, y comprobar que la nueva DTD Reducida se comportaba bien, obtuvimos el siguiente resultado:

Composición de la DTD Reducida:

rdocbookx.dtd (../dtd/docbook/xml/reducida/4.1.2.1/rdocbookx.dtd)

La DTD DocBook XML Reducida

rdbpoolx.mod (../dtd/docbook/xml/reducida/4.1.2.1/rdbpoolx.mod)

El módulo "piscina de información" de la DTD DocBook XML Reducida

rdbhierx.mod (../dtd/docbook/xml/reducida/4.1.2.1/rdbhierx.mod)

El módulo "organización" de la DTD DocBook XML Reducida

rdbnotnx.mod (../dtd/docbook/xml/reducida/4.1.2.1/rdbnotnx.mod)

El módulo "notaciones" de la DTD DocBook XML Reducida

rdbcentx.mod (../dtd/docbook/xml/reducida/4.1.2.1/rdbcentx.mod)

El módulo "entidades para los caracteres" de la DTD DocBook XML Reducida

rcalstblx.dtd (../dtd/docbook/xml/reducida/4.1.2.1/rcalstblx.dtd)

Versión XML del modelo de Tablas CALS SGML

rsoextblx.dtd (../dtd/docbook/xml/reducida/4.1.2.1/rsoextblx.dtd)

El modelo XML de Intercambio de Tablas (http://www.oasis-open.org/html/tm9901.htm) de OASIS Open (http://www.oasis-open.org/). Esta es el modelo de tablas alternativo para la versión XML de DocBook Reducida

rdocbook.cat (../dtd/docbook/xml/reducida/4.1.2.1/rdocbook.cat)

Un catálogo para la DTD DocBook XML Reducida

rinclx.mod (../dtd/docbook/xml/reducida/4.1.2.1/rinclx.mod)

Conjunto de entidades eliminadas de la DTD DocBook XML v4.1.2

Bibliografía

La documentación consultada para poder llevar a cabo este proyecto se detalla a continuación:

http://www.docbook.org

El libro DocBook de O'Reilly. Sobre todo el capítulo 5 - Customizing DocBook (http://docbook.org/tdg/en/html/ch05.html)

http://www.oasis-open.org/docbook/xml/4.1.2/index.shtml

La página de Oasis-Open (http://www.oasis-open.org/) donde se encuentra la DTD DocBook XML v4.1.2

http://www.oasis-open.org/docbook/xml/simple/4.1.2.5/index.shtml

La página de Oasis-Open (http://www.oasis-open.org/) donde se encuentra la DTD Simplified DocBook XML v4.1.2.5

http://es.tldp.org/Tutoriales/DOCBOOK/multiple-html/

El tutorial de Jaime Irving Dávila sobre DocBook

http://es.tldp.org/Manuales-LuCAS/FLEX/flex-es-2.5.html

La página del manual de flex

http://perldoc.com

La página de documentación de Perl

http://www.comp.leeds.ac.uk/Perl/start.html

Un tutorial sobre Perl

Sobre este documento

Este documento ha sido escrito en formato XML utilizando la DTD DocBook XML Reducida v4.1.2.1, obtenida como resultado del presente proyecto. Esta DTD es una versión simplificada de la DTD DocBook XML 4.1.2 (http://www.oasis-open.org/docbook/xml/4.1.2/index.shtml).

A. Analizador de etiquetas

Para obtener las etiquetas empleadas en un determinado documento escrito en DocBook XML, se generó el programa analizador_tags. Este programa, escrito en flex (ftp://ftp.gnu.org/pub/non-gnu/flex/index.html)[1], acepta como parámetros los documentos[2] de los cuales queremos obtener las etiquetas más usadas. Si no se le pasa ningún parámetro, toma los datos de la entrada estándar.

Este programa analiza los documentos pasados como parámetros de la siguiente forma:

1. Si es un comentario, lo ignora. Un comentario es todo aquello comprendido entre: "<!--" y "-->".
2. Si es la definición de documento XML, la ignora. Esta definición está comprendida entre: "<?xml" y "?>".
3. Si es una definición del tipo de documento, la ignora. Esta definición está delimitada por: "<!DOCTYPE" y ">"
4. Si se encuentra la declaración de una entidad, la ignora. Una entidad está comprendida entre: "<!ENTITY" y ">"

5. Ignoramos todos los caracteres comprendidos entre: "<![CDATA[" y]]">".
6. Ignoramos las etiquetas de cierre: "</etiqueta>".
7. Ignoramos todo carácter comprendido entre: "<?" y "?>".
8. Si encuentra un carácter "<" (que no es un caso especial), quiere decir que estamos ante una nueva etiqueta. El nombre de una etiqueta está comprendido entre "<" y el primer espacio en blanco, tabulación o final de etiqueta "/" que se encuentre. Todo lo que venga después del nombre de una etiqueta se ignora, hasta encontrar el carácter de cierre de etiqueta: ">".

   Ejemplos de etiquetas: <para>, <imagedata/>, <sect1 id="index">. En estos casos, las etiquetas obtenidas serían: "para", "imagedata" y "sect1".

9. Todo carácter que no se encuentre en cualquiera de estos caso, como pueden ser los finales de línea y el contenido del documento, no se tienen en cuenta.

Obtención del programa ejecutable

En el directorio codigo (../codigo/codigo.html), se encuentra el archivo original "analizador_tags.l (../codigo/analizador_tags.l)". Para obtener el programa ejecutable, primero tenemos que obtener el código fuente en C, para lo cual teclearemos:

```
[fyr@todoscsi]$ flex analizador_tags.l
```

Tras lo cual obtendremos el archivo `lex.yy.c`, que tendremos que compilar con un compilador de C y enlazarlo con la librería -lfl. Para ello teclearemos:

```
[fyr@todoscsi]$ gcc lex.yy.c -lfl
```

Una vez finalizada la compilación, obtendremos un ejecutable, a.out.

Análisis léxico de ejemplo

En estos momentos ya tenemos el analizador léxico disponible, el cual podremos invocar como se muestra a continuación:

```
[fyr@todoscsi]$ ./a.out archivo.xml
```

La ejecución del programa nos devolverá las etiquetas según las vaya encontrando, una por línea. Así, una salida del programa, para el siguiente archivo:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                  "file:///usr/share/sgml/docbook/dtd/xml/4.1.2/docbookx.dtd">
<article id="article">
<articleinfo>
<title>Unit Test: article.001.xml</title>
<releaseinfo role="CVS">$Id: reduccion_dtd_docbook.xml,v
 1.1 2002/09/09 20:57:14 sergio Exp $</releaseinfo>
<authorgroup>
```

```
<author><firstname>Norman</firstname><surname>Walsh</surname>
<affiliation><address><email>ndw@nwalsh.com</email></address></affiliation>
</author>
<author><firstname>Jane</firstname><surname>Doe</surname></author>
</authorgroup>
<abstract>
<para>This is the abstract.</para>
<para>It has several paras.</para>
<para>It has several paras.</para>
</abstract>
</articleinfo>

<para>This is an article tests.</para>

<ackno>I'd like to thank all the tests that came before me.</ackno>

</article>
```

Sería:

```
[fyr@todoscsi]$ ./a.out archivo.xml

article
articleinfo
title
releaseinfo
authorgroup
author
firstname
surname
affiliation
address
email
author
firstname
surname
abstract
para
para
para
para
ackno

[fyr@todoscsi]$
```

Código fuente del analizador léxico

El siguiente código se corresponde con el analizador léxico creado con flex:

```
/*
 * analizador_tags.l - Programa escrito en Flex que devuelve las etiquetas
```

```
 *                     contenidas en los archivos pasados como parámetros,
 *                     o en el texto obtenido de la entrada estándar.
 *
 *                     Este programa está pensado para analizar documentos
 *                     escritos en DocBook XML v4.1.2 bien formados.
 *
 *
 * Copyright (C) 2002
 *     Fernando Reyero Noya     <fernando.reyero@hispalinux.es>
 *     Sergio González González <sergio.gonzalez@hispalinux.es>
 *
 *
 *  This program is free software; you can redistribute it and/or modify
 *  it under the terms of the GNU General Public License as published by
 *  the Free Software Foundation; either version 2, or (at your option)
 *  any later version.
 *
 *  This program is distributed in the hope that it will be useful,
 *  but WITHOUT ANY WARRANTY; without even the implied warranty of
 *  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *  GNU General Public License for more details.
 *
 *  You should have received a copy of the GNU General Public License
 *  along with this program; if not, write to the Free Software Foundation,
 *  Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
 *
 */


%x comentario
%x definicionXML
%x definicionDTD
%x ENTITY
%x CDATA
%x cierre_etiqueta
%x especiales
%x etiquetas

int etiqueta_encontrada = 0;

%%


"<!--" BEGIN ( comentario ) ;
"<?xml" BEGIN ( definicionXML ) ;
"<!DOCTYPE" BEGIN ( definicionDTD ) ;
"<!ENTITY" BEGIN ( ENTITY ) ;
"<![CDATA[" BEGIN ( CDATA ) ;
"</" BEGIN ( cierre_etiqueta ) ;
"<?" BEGIN ( especiales ) ;
"<" BEGIN ( etiquetas ) ;
.|\n ;
```

```
<comentario>{

/*
 * Ignoramos los comentarios. Un comentario es de la forma:
 *
 *         <!--   contenido del comentario -->
 */

[^\-\n]*         /* ignora todo lo que no sea un guión: '-' */
"-"+[^("->")\n]* /* ignora todos los guiones '-' que no vayan seguidos de un '>' */
\n               /* ignora los saltos de línea */
\-{2,}+>         BEGIN ( INITIAL ); /* Hemos llegado al final del comentario: '-->' */
}



<definicionXML>{

/*
 * Ignoramos la definición de documento XML. Esta definición es de la forma:
 *
 *        <?xml version="1.0" encoding="ISO-8859-1" ?>
 */

[^?\n]*          /* ignora todo lo que no sea un interrogante: '?' */
"?"+[^("?>")\n]* /* ignora todos las '?' que no vayan seguidas de un '>' */
\n               /* ignora los saltos de línea */
\?+>             BEGIN ( INITIAL ); /* Hemos llegado al final de la definición
                                     * de documento XML
                                     */
}



<definicionDTD>{

/*
 * Ignoramos la definición de tipo de documento. Una definición de tipo de documento
 * es de la forma:
 *
 *   <!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
 *                 "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
 */

[^>\n]*  /* ignora todo lo que no sea '>' */
\n       /* ignora los saltos de línea */
>        BEGIN ( INITIAL ); /* Hemos llegado al final de la definición de tipo
                             * de documento
                             */
}
```

```
<ENTITY>{

/*
 * Ignoramos las definición de entidades, si las hubiera. Una entidad se define de
 * la siguiente forma:
 *
 *   <!ENTITY capitulo0 SYSTEM "capitulo0.xml">
 */

[^>\n]*  /* ignora todo lo que no sea '>' */
\n       /* ignora los saltos de línea */
>        BEGIN ( INITIAL ); /* Hemos llegado al final de la definición de
                               * tipo de documento
                               */

}


<CDATA>{

/* Ignoramos los caracteres comprendidos entre "<![CDATA["  y  "\]\]>" */

[^\]\n]*          /* ignora todo lo que no sea un corchete: ']' */
\]+[^(\]>)\n]*    /* ignora todos los corchetes ']' que no vayan seguidos de un '>' */
\n                /* ignora los saltos de línea */
\]{2,}+>          BEGIN ( INITIAL ); /* Hemos llegado al final de la definición
                                        * de CDATA
                                        */
}



<cierre_etiqueta>{

/* Ignoramos los cierres de etiqueta */

[^>\n]*  /* ignora todo lo que no sea '>' */
\n       /* ignora los saltos de línea */
>        BEGIN ( INITIAL ); /* Hemos llegado al final del cierre de la etiqueta */
}



<especiales>{

/*
 * Ignoramos las entradas especiales del tipo:
 *
 *   <? ... ?>
 */
```

```
[^\?\n]*        /* ignora todo lo que no sea '?' */
\?+[^(\?>)\n]* /* ignora todos los interrogantes '?' que no vayan seguidos de un '>' */
\n              /* ignora los saltos de línea */
\?+>            BEGIN ( INITIAL ); /* Hemos llegado al final de la definición de
                                    * tipo de documento
                                    */
}



<etiquetas>{

/*
 * Obtenemos y contamos las etiquetas del documento. Una etiqueta está delimitada
 * por un '<' en su comienzo y por un '>' o un espacio en blanco en su final.
 *
 * Ejemplo1: <para>, en este caso, 'para' sería la etiqueta
 * Ejemplo2: <sect1 id="index">, en este caso, 'sect1' sería la etiqueta, todo lo
 *           que viene después del espacio en blanco se ignora, hasta el carácter '>'
 */

[^(> \t\/)\n]* {
/* muestra todo lo que no sea un '>', un espacio en blanco o "/" */
printf("%s\n", yytext);
etiqueta_encontrada = 1;
}
\n               /* ignora los saltos de línea */
>|[ \t]+|\/+|\n.    BEGIN ( INITIAL ); /* Hemos llegado al final de la etiqueta */
}



%%

int contador;

main( argc, argv )
int argc;
char **argv;
{
++argv, --argc; /* ignoramos el nombre del programa */

if ( argc > 0 ) {
/* Se han tecleado parámetros, los analizamos */

while ( argc > 0 ) {

if ( (yyin = fopen( argv[0], "r" )) == NULL ) {

fprintf(stderr,
"\n** Error al abrir el fichero \"%s\", pasamos al siguiente archivo... **\n\n",
argv[0]);
++argv, --argc; /* obtenemos el siguiente parámetro */
```

```
continue;

} else {

yylex();
++argv, --argc; /* obtenemos el siguiente parámetro */

} /* else */

} /* while() */

} else {

/* No se han tecleado parámetros, tomamos los datos de la entrada estándar */
yyin = stdin;
yylex();

} /* else */
}/* main */
```

B. Contador de etiquetas

Para contar las etiquetas obtenidas por el analizador léxico[1], generamos el script en perl denominado: "contador_tags.pl (../codigo/contador_tags.pl)", almacenado en el directorio codigo (../codigo/).

Este script acepta como parámetro la ruta hacia los documentos escritos en DocBook XML que queremos analizar. Si no se le pasa ningún parámetro, obtiene los documentos de ../documentos/docbook_xml/.

Este script realiza los siguientes pasos:

1. Se obtiene la lista de documentos para su posterior análisis (dependiendo del parámetro pasado, se obtendrán de un lugar u otro).
2. Se analiza cada uno de los documentos buscando posibles errores en el código XML[2]. Si se encuentran errores en un documento, se notifica al usuario, y dicho documento no será tratado por el analizador léxico.
3. Una vez comprobada la correcta formación de los documentos, se buscan las etiquetas de aquellos documentos correctamente formados[3]. Las etiquetas se obtienen invocando al analizador léxico descrito en el Apéndice A.
4. Llegados a este punto, el script cuenta las etiquetas y muestra el resultado por pantalla[4].

Un ejemplo

Para ver como funciona este script, veamos un ejemplo. Vamos a analizar los documentos contenidos en el directorio desde donde se ejecuta el script. Supongamos que en dicho directorio sólo se encuentra un fichero XML con el siguiente contenido:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                  "file:///usr/share/sgml/docbook/dtd/xml/4.1.2/docbookx.dtd">
<article id="article">
<articleinfo>
<title>Unit Test: article.001.xml</title>
<releaseinfo role="CVS">$Id: reduccion_dtd_docbook.xml,v 1.1
2002/09/09 20:57:14 sergio Exp $</releaseinfo>
<authorgroup>
<author><firstname>Norman</firstname><surname>Walsh</surname>
<affiliation><address><email>ndw@nwalsh.com</email></address></affiliation>
</author>
<author><firstname>Jane</firstname><surname>Doe</surname></author>
</authorgroup>
<abstract>
<para>This is the abstract.</para>
<para>It has several paras.</para>
<para>It has several paras.</para>
</abstract>
</articleinfo>

<para>This is an article tests.</para>

<ackno>I'd like to thank all the tests that came before me.</ackno>

</article>
```

Al ejecutar el script sobre este directorio, obtendremos la siguiente salida:

```
[fys@todoscsi]$ ./contador_tags.pl ./


Buscando los documentos... [Hecho]

Analizando la validez de los documentos... [Hecho]

Buscando etiquetas... [Hecho]

Contando las etiquetas... [Hecho]

A continuación se mostrarán las etiquetas y el número de apariciones:

Número                  Etiquetas
apariciones

2                       firstname
1                       ackno
4                       para
2                       author
2                       surname
1                       authorgroup
1                       email
1                       article
```

```
1                       affiliation
1                       title
1                       articleinfo
1                       address
1                       abstract
1                       releaseinfo
[fys@todoscsi]$
```

Código fuente del contador de etiquetas

El siguiente código se corresponde con el contador de etiquetas creado en perl:

```
#!/usr/bin/perl
#
#   contador_tags.pl - Script que obtiene un reporte sobre las etiquetas
#                       más utilizadas en los documentos, escritos en
#                       DocBook XML, analizados.
#
#
#  NOTA: para la correcta ejecución de este script, es necesario
#        tener instalado el analizador gramatical de XML "rxp".
#        Indicad en la definición de variables, la localización
#        de este programa en vuestro sistema.
#
#  http://www.cogsci.ed.ac.uk/~richard/rxp.html
#
#
#   Copyright (C) 2002
#
#      Fernando Reyero Noya <fernando.reyero@hispalinux.es>
#      Sergio González González <sergio.gonzalez@hispalinux.es>
#
#
#   This program is free software; you can redistribute it and/or modify
#   it under the terms of the GNU General Public License as published by
#   the Free Software Foundation; either version 2, or (at your option)
#   any later version.
#
#   This program is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warranty of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#   GNU General Public License for more details.
#
#   You should have received a copy of the GNU General Public License
#   along with this program; if not, write to the Free Software Foundation,
#   Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
#


# -------------------------------
# Definición de algunas variables
```

```
#

# Directorio por defecto donde se encuentran los archivos a analizar
my $directorio_ejemplos = "../documentos/docbook_xml_reducida/";

# Extensión de los archivos a analizar
my $extension = "xml";


# Array que almacena los posibles archivos a analizar
my @posibles_archivos_analizar;

# Array que almacena los archivos verificados que serán finalmente
# analizados
my @archivos_analizar_verificados;

# Array que almacena todas las etiquetas obtenidas de los documentos analizados
my @todas_las_etiquetas;

# Array asociativo que almacena el par "nombre_de_la_etiqueta, número_de_apariciones"
my %apariciones_etiquetas;


# Localización de 'rxp'
my $rxp = "/usr/bin/rxp";

# Localización de 'find'
my $find = "/usr/bin/find";

# Localización del analizador de etiquetas
my $analizador_etiquetas = "./analizador_tags";



# -------------------
# Comienzo del script
#


#
# Verificamos que los programas necesarios para ejecutar este script
# están presentes en el sistema.
#

unless (-e $rxp)
{
# rxp no se encuentra en la ruta especificada, salimos del programa
print "\n * $! '$rxp' (programa abortado)\n\n";
exit -1;
}

unless (-e $analizador_etiquetas)
{
```

```
# El analizador de etiquetas no se encuentra en la ruta especificada,
# salimos del programa
print "\n * $! '$analizador_etiquetas' (programa abortado)\n\n";
exit -1;
}


#
# Analizamos los parámetros pasados al script:
#
#    - Si no se pasa ningún parámetro, se obtienen los archivos de la
#      ruta por defecto.
#
#    - Este script acepta un parámetro: la ruta donde están almacenados
#      los documentos a analizar.
#


$directorio_ejemplos = $ARGV[0] if( length($ARGV[0]) != 0 );

# comprobamos si existe el directorio de archivos a analizar
unless (-d $directorio_ejemplos)
{
print "\n * $!: '$directorio_ejemplos' *\n\n";
exit -1;
}

#
# Obtenemos los archivos que debemos analizar
#

print "\nBuscando los documentos... ";

@posibles_archivos_analizar =
             `$find $directorio_ejemplos -name "*.$extension" 2>errores.log`;

#
# Comprobamos la salida de find. Si ha tenido algún error, lo notificamos y
# salimos del script
#

if (($? >> 8) > 0)
{
print "\n\n ERROR al ejecutar `find \$directorio_ejemplos -name \"*.\$estension\"`
                                            (compruebe el archivo errores.log)\n\n";
exit -1;
}
else
{
if (-e "errores.log") { `rm -rf errores.log`; }
}

print "[Hecho]\n\n"; # Búsqueda de documentos
```

```
#
# Analizamos el documento generado a partir de las porciones (si está presente
# el programa rxp en el sistema), para ver si contiene errores.
#

print "Analizando la validez de los documentos... ";

my $error = 0; # Variable que indica si hemos tenido algún error
               # en el análisis de los documentos

foreach $documento ( @posibles_archivos_analizar )
{
chomp $documento;
`echo "\n########## $documento ##########\n" >> rxp.log`;
`$rxp -VVNx $documento >/dev/null 2>>rxp.log`;

if ( (($? >> 8) == 0) )
{
# Eliminamos el archivo rxp.log
if ( (-e "rxp.log") && !$error ) { `rm -rf rxp.log`; }
push(@archivos_analizar_verificados, $documento);

}
else
{
# Alguna porción está mal formada o posee errores
print"\n ** Aviso ** los siguientes archivos contienen errores y no se
                                     tendrán en cuenta: \n\n" if ( !$error );
print "  - error en el documento: '$documento'\n";
$error = 1;
} # else
} # foreach

print "\nRevisa el archivo rxp.log para ver los errores...\n\n" if ( $error );

print "[Hecho]\n\n";


#
# Analizamos los documentos válidos en busca de sus etiquetas y las almacenamos
# en el array: "todas_las_etiquetas"
#

print "Buscando etiquetas... ";

foreach $archivo (@archivos_analizar_verificados)
{
@todas_las_etiquetas = (@todas_las_etiquetas, `$analizador_etiquetas $archivo`);
}

print "[Hecho]\n\n";
```

```
#
# Contamos el número de etiquetas
#

print "Contando las etiquetas... ";

foreach $etiqueta ( @todas_las_etiquetas )
{
chomp $etiqueta;
if ( exists $apariciones_etiquetas{ $etiqueta } )
{
# Sumamos una unidad a la etiqueta
%apariciones_etiquetas = (%apariciones_etiquetas, $etiqueta,
                                            $apariciones_etiquetas{"$etiqueta"}+1);
}
else
{
%apariciones_etiquetas = (%apariciones_etiquetas, $etiqueta, 1);
}
} # foreach

print "[Hecho]\n\n";


#
# Mostramos los resultados obtenidos
#

print "A continuación se mostrarán las etiquetas y el número de apariciones:\n\n";
print "Número\t\t\tEtiquetas\napariciones\n\n";

while( ($etiqueta, $numero_apariciones) = each(%apariciones_etiquetas) )
{
print "$numero_apariciones\t\t\t$etiqueta\n";
}
```

C. Generación de la documentación

Gracias al uso de la DTD DocBook, podremos obtener fácilmente distintos formatos de esta documentación. Para facilitar el proceso, hemos creado el script genera.sh (./genera.sh) que se encarga de generar los siguientes formatos, a partir del código en XML (./reduccion_dtd_docbook.xml) de esta documentación:

- HTML en varias páginas (./index.html)
- HTML en un bloque (./reduccion_dtd_docbook.html)
- TeX (./reduccion_dtd_docbook.tex)

- DVI (./reduccion_dtd_docbook.dvi)
- PDF (./reduccion_dtd_docbook.pdf)
- PostScript (./reduccion_dtd_docbook.ps)
- RTF (./reduccion_dtd_docbook.rtf)

Para generar los distintos formatos sólo tenemos que teclear lo siguiente el el directorio documentacion (./):

```
[fys@todoscsi]$ ./genera.sh reduccion_dtd_docbook
```

> **Aviso**
>
> Nótese que no se ha incluido la extensión al archivo

Tras lo cual obtendremos:

```
[fys@todoscsi]$ ./genera.sh reduccion_dtd_docbook


Comenzando la generación de la documentación...


Generando HTML simple...

  [Hecho]

Generando HTML en partes...

  [Hecho]

Generando archivo .tex...

  [Hecho]

Generando DVI...

  [Hecho]

Generando PDF...

  [Hecho]

Generando PostScript...

  [Hecho]

Generando RTF...

  [Hecho]

Documentación generada.
```

```
[fys@todoscsi]$
```

Una vez ha finalizado el script, ya disponemos de los distintos formatos en el directorio documentacion (./).

Código del generador

El código empleado para generar la documentación se puede observar a continuación:

```
#!/bin/bash
#
# Copyright (C) 2002
#     Fernando Reyero Noya     <fernando.reyero@hispalinux.es>
#     Sergio González González <sergio.gonzalez@hispalinux.es>#
#
#
#  This program is free software; you can redistribute it and/or modify
#  it under the terms of the GNU General Public License as published by
#  the Free Software Foundation; either version 2, or (at your option)
#  any later version.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License
#  along with this program; if not, write to the Free Software Foundation,
#  Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
#

# Localización de los binarios
JADE=/usr/bin/jade
XSLTPROC=/usr/bin/xsltproc
JADETEX=/usr/bin/jadetex
DVIPDF=/usr/bin/dvipdf
DVIPS=/usr/bin/dvips

# Hoja de estilo html normal (todo en uno)
ESTILO_HTML=./fys.xsl

# Hoja de estilo html en trozos
ESTILO_HTML_CHUNK=./fys-chunk.xsl

# Hoja de estilo dssl (para pasar a .tex)
ESTILO_TEX=/usr/share/sgml/docbook/stylesheet/dsssl/modular/print/docbook.dsl

# Declaración de entidades
ENT=/usr/share/sgml/declaration/xml.dcl
```

```
/bin/echo -e "\nComenzando la generación de la documentación...\n\n"

# Pasamos a HTML normal
/bin/echo -e "Generando HTML simple...\n"
$XSLTPROC $ESTILO_HTML "$1".xml > "$1".html 2>/dev/null
/bin/echo -e "  [Hecho]\n"

# Pasamos a HTML en trozos
/bin/echo -e "Generando HTML en partes...\n"
$XSLTPROC $ESTILO_HTML_CHUNK "$1".xml 2>/dev/null
/bin/echo -e "  [Hecho]\n"

# Pasamos a formato TEX
/bin/echo -e "Generando archivo .tex...\n"
$JADE -d $ESTILO_TEX -t tex -V tex-backend $ENT "$1".xml 2>/dev/null
/bin/echo -e "  [Hecho]\n"

# Pasamos a DVI
/bin/echo -e "Generando DVI...\n"
$JADETEX "$1".tex >/dev/null 2>/dev/null
/bin/echo -e "  [Hecho]\n"

# Pasamos de DVI a PDF
/bin/echo -e "Generando PDF...\n"
$DVIPDF "$1".dvi 2>/dev/null
/bin/echo -e "  [Hecho]\n"

# Pasamos a de DVI a PostScript
/bin/echo -e "Generando PostScript...\n"
$DVIPS "$1".dvi 2>/dev/null
/bin/echo -e "  [Hecho]\n"

# Pasamos a RTF
/bin/echo -e "Generando RTF...\n"
$JADE -d $ESTILO_TEX -t rtf $ENT "$1".xml 2>/dev/null
/bin/echo -e "  [Hecho]\n"

# Borramos los ficheros temporales generados
/bin/rm -f *.aux *.log *.out
/bin/echo -e "Documentación generada.\n"
```

Notas

1. Los documentos analizados se han obtenido de diversos lugares de Internet, como puede apreciarse en la bibliografía.
2. Esta duplicación se ha realizado para comprobar el correcto funcionamiento de la DTD Reducida.
3. consultar el Apéndice A para saber como funciona
4. para más información lee el Apéndice B.

5. Para ello hace uso del analizador rxp (http://www.cogsci.ed.ac.uk/~richard/rxp.html).
6. Esto se logra gracias al analizador léxico, ya que una vez analizados los documentos que le pasa en script en perl, este le devuelve las etiquetas que ha encontrado.
7. Esta es la ruta por defecto que toma el script “contador_tags.pl”, si no se le pasa ningún parámetro.

1. Se ha empleado flex para esta parte del análisis debido a la rapidez y facilidad que se crean analizadores léxicos con esta herramienta.
2. El analizador léxico da por supuesto que los documentos analizados están correctamente escrito, por lo que no hace ninguna comprobación de error (de esto se encarga el contador de etiquetas - ver Apéndice B -).

1. ver Apéndice A
2. Para la verificación de los documentos XML se ha utilizado la herramienta rxp (http://www.cogsci.ed.ac.uk/~richard/rxp.html).
3. Las etiquetas son almacenadas en un array para su posterior análisis.
4. Se recomienda hacer uso de herramientas como sort, para ordenar la salida del programa según sus preferencias.