



DOCUMENTOS XML

Jaime E. Villate
Universidad de Oporto.

Este artículo explica cómo escribir documentos y páginas para publicar en la *web*, usando *LE-document*, un tipo de documento basado en la versión XML de *DocBook*.

Usando una versión simplificada de *DocBook* se facilita el proceso de creación de documentos, y el editor de texto puede dar mayor ayuda en la construcción de la estructura del documento. Para transformar los ficheros fuente XML para otros formatos, se muestran algunos ejemplos de páginas de estilo XSL. Los ejemplos que se dan han sido creados en un sistema *Debian*, pero como no se usa ningún programa específico de *Debian*, deberá funcionar igual en cualquier distribución de *Linux*, una vez sean instalados los programas necesarios.

El sistema de creación de páginas *web* y manuales que se describe en este artículo ha sido creado para el proyecto *La Espiral* (<http://laespiral.org>).

Este proyecto busca impulsar el uso del *software libre*, y en particular de *Debian GNU/Linux*, por parte de usuarios de habla hispana sin conexión permanente a *Internet*. Este tipo de usuarios tiene como principal fuente de instalación de *Linux* los discos compactos que se distribuyen con revistas. Para facilitar su distribución, *La Espiral* va a crear selecciones de *software* basadas en *Debian* y acompañadas de documentación que pueda ser distribuida libremente.

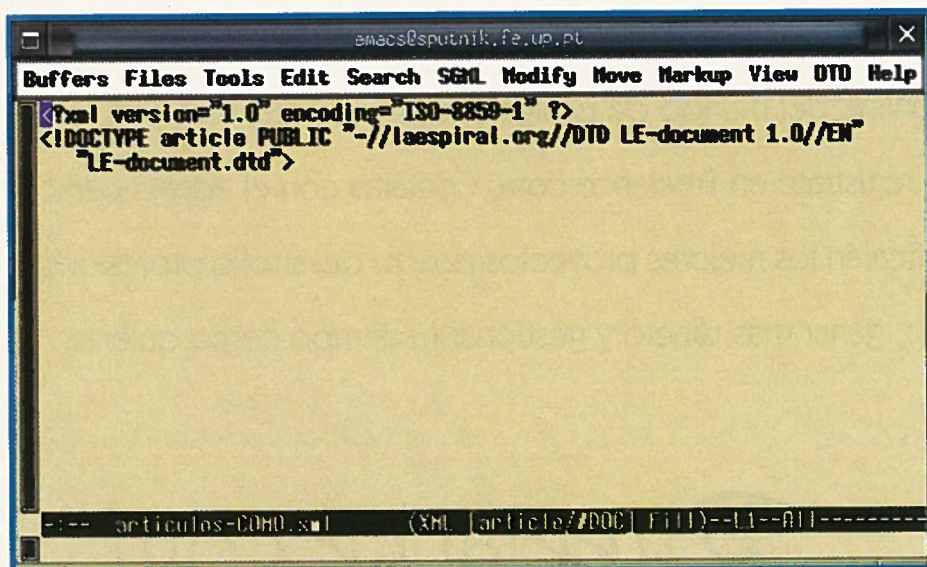
Para cumplir este objetivo, los documentos distribuidos por el proyecto deberán estar disponibles en versiones para imprimir con buena calidad, pero al mismo tiempo deberán existir versiones destinadas a consulta rápida en la *web*. Otro re-

‘El mismo método usado para los artículos se puede usar para producir páginas web’

quisito importante es mantener un estilo uniforme, ya que el proyecto se basa en contribuciones de muchos voluntarios.

La solución que proponemos aquí, bautizada como *LE-document*, emplea XML para los ficheros fuente y unas páginas de estilo XSL para producir los diferentes formatos. Esta solución presenta varias ventajas:

- Permite definir el documento de una forma estructurada.
- Facilita la producción de varios formatos diferentes (*html*, *latex*, *pdf*, *postscript*, texto en *ascii*)
- Hace posible la extracción de partes de un documento para introducirlas automáticamente en una base de datos.
- Permite definir un estilo uniforme para todos los artículos del proyecto, el cual puede ser modificado en cualquier momento, sin tener que modificar ninguno de los artículos.
- Permite la construcción de todas las páginas *web* del proyecto usando la misma herramienta usada para los artículos.



1 Comienzo de un artículo usando *LE-Document.dtd*.





‘El modo xml se encarga de leer la línea que define el tipo de documento’

Un documento escrito en XML puede parecer complicado por la cantidad de etiquetas y atributos que es necesario introducir, pero con un editor de texto adecuado y un DTD (fichero que define la estructura del documento) bien diseñado, las etiquetas podrán ser introducidas fácilmente y el propio editor podrá sugerir la secuencia lógica que deben seguir las etiquetas. Antes de dar un ejemplo de cómo escribir un artículo, vamos a sugerir los programas que deben ser instalados para facilitar el proceso.

El DTD que usaremos es una versión simplificada de *DocBook*, que es el sistema preferido por muchos proyectos de documentación libre. La ventaja de usar una versión simplificada, en vez del original, reside en que el proceso de creación de un documento se facilita mucho más, al igual que la elaboración de páginas *xsl* para transformar el código *xml* en otros formatos. Cualquier documento creado usando el sistema que se describe en este artículo puede ser usado igualmente en *DocBook*, bastando sustituir la definición del DTD al comienzo del fichero.

REQUISITOS

Para facilitar la redacción de artículos usando *LE-document* en *Debian*, es aconsejable instalar los siguientes paquetes:

Emacs. Combinado con un modo para edición de ficheros XML, facilita el trabajo de escribir un artículo en XML.

Psgml. Modo para editar ficheros *sgml/html* en *emacs*. Las versiones más recientes (1.2.1-8 o superior) admiten mejor el código XML.

Sp. Incluye el programa *nsgmls* que permite examinar la sintaxis de un fichero XML.

Sablotron. Procesador de páginas XSL (lenguaje de descripción de estilo), útil para transformar la fuente *xml* de un artículo en *html* o *latex*, usando páginas de descripción de estilo. Se recomienda usar la versión 0.43 o superior. En versiones anteriores el resultado sale en código UFT8, siendo necesario usar un programa, por ejemplo *Recode*, para transformarlo a código *iso-latin-1*. Los paquetes de *sablotron* y *libsablot0* en *Woody* pueden ser instalados sobre *Potato*, y los puede copiar de <http://laespiral.org/software/>.

Tetes. Para producir las versiones *PostScript* y *pdf*, usaremos un fichero *LaTeX* generado a partir del fichero *xml*. En *Debian* será necesario instalar los paquetes *tetex-base* y *tetex-extra*. Usaremos también el paquete *hyperref*, que es parte de *tetex-extra*; sin embargo, las versiones actuales de este paquete en *Debian* no funcionan en español, ya que no reconocen la ñ (aparece un mensaje de error: *Command \textitilde unavailable in encoding OT1*). Este problema sólo fue corregido muy recientemente; para actualizarse a la versión de *hyperref* que funciona en español, basta copiar los ficheros que se encuentran dentro del archivo <http://laespiral.org/software/hyperref.6.70f.tgz>; esto no creará ningún conflicto con el paquete *tetex-extra* original de *Debian* que ya ha sido instalado.

le-docxml. Este es un paquete no oficial de *Debian*, que podemos encontrar en <http://laespiral.org/software/>

(si trabaja con otra distribución diferente de *Debian*, deberá copiar el archivo *le-docxml-1.0.tar.gz* y seguir las instrucciones para su instalación). Este paquete contiene el fichero de definición del tipo de documento que vamos a crear, así como las páginas de estilo y programas para producir las versiones *html* y *latex* a partir de la fuente *xml*.

PRIMEROS PASOS

Para comenzar a escribir un artículo, el primer paso es crear un fichero con extensión *xml* con el siguiente contenido en las dos primeras líneas:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE article PUBLIC "-//laespiral.org/DTD
LE-document 1.0//EN"
"LE-document.dtd">
```

La primera línea indica que lo que viene a continuación es un fichero en código *xml*, versión 1.0, y con caracteres del código *ISO-8859-1*. La segunda línea define el DTD (en este caso *LE-document.dtd*) y da alguna información sobre la institución que lo ha producido, el nombre oficial que se le ha dado, y el idioma en que fue escrito. Un DTD define un árbol jerárquico de elementos; dentro de la línea donde se identifica el DTD que será usado, debe identificarse también cual será el elemento principal, dentro de esa jerarquía, que será usado en el presente documento. En este caso hemos escrito

2 Estructura mínima de un artículo en *LE-Document.dtd*.

article, pues vamos a escribir un artículo completo. Si fuéramos a escribir únicamente una sección de un artículo, escribiríamos, por ejemplo, *sect1*, o si fuéramos a crear una página para la web, escribiríamos *webpage*.

Si se abre con *emacs* un fichero con extensión *xml* que tenga las dos líneas anteriores, el resultado será como el de la Figura 1.

La extensión *xml* del nombre del fichero ha hecho que *emacs* ejecute el modo XML de *psgml*, como se puede ver en el centro de la línea de estado (la línea negra en la parte inferior) y por la aparición de varios menús adicionales para trabajar con *xml*. El modo *xml* se ha encargado también de leer la línea que define el tipo de documento y ha cargado el fichero *LE-document.dtd* que se encuentra en el directorio de trabajo (si está en otro directorio, la línea de definición del documento debe ser modificada dando el camino completo del fichero *dtd*).

En la línea de estado en la Figura 1 se puede ver que el fichero *dtd* ya ha sido leído y analizado, pues ya ha sido identificado el elemento principal del documento: *article*. También puede ver alguna información sobre el DTD y los elementos que define, en el menú DTD que presenta *psgml* en *emacs*. Si en su sesión de *emacs* no aparece esa información, por ejemplo si comenzó a escribir las dos primeras líneas en un fichero vacío, tendrá primero que asegurarse que está en modo XML, con el comando *M-x xml-mode*, y después podrá seleccionar la opción *Parse DTD* en el menú DTD (o si prefiere usar el teclado: *C-c C-p*).

ESCRIBIENDO CON LE-DOCUMENT

Después de estar en modo XML y de haber seleccionado un DTD, puede usar una opción de menú muy útil, que será la que nos permitirá escribir el documento rápidamente; se trata de la opción *Insert Element*, en el menú *Markup*. Esta opción nos da una lista de los elementos que son permitidos en el punto donde se encuentra el cursor. Escogiendo un elemento en la lista, son introducidas las etiquetas requeridas por ese elemento, y si existe alguna in-

formación adicional obligatoria, será pedida en el *mini-buffer* (la última línea en la pantalla).

Otra forma rápida de introducir elementos, equivalente a la opción *Insert Element*, consiste en oprimir simultáneamente la tecla *Shift* y el botón derecho del ratón. Al hacer eso, aparece un cuadro con la lista de elementos válidos; por ejemplo

en la Figura 1 se acabó de pulsar *Shift+botón derecho* y ha aparecido una lista con un único elemento, *article*, que es el único elemento que se puede seleccionar inicialmente. Seleccionando este elemento, aparece toda la información que se muestra en la Figura 2.

El contenido del fichero en la Figura 2 constituye la estructura mínima que debe tener un artículo en el sistema *LE-article*. El cursor ha sido desplazado al primer lugar donde tendremos que comenzar a introducir texto: el título del artículo. Después de escribir el título del artículo, podemos usar la opción *Next data field*, en el menú *Move* (o con el teclado: *C-c C-d*), para desplazarnos al próximo campo que debe ser rellenado.

Repetiendo el comando *Next data field*, verá que inicialmente los únicos campos que tiene para rellenar son el título del artículo, el nombre y apellidos del autor (*firstname* y *surname*) y el título de la primera sección. El resto del artículo es considerado opcional en el DTD que se está usando. Para introducir otra información opcional, conviene revisar qué otros elementos y atributos opcionales existen en cada sección. Por ejemplo después del final del elemento *author* y antes de terminar *artheader*, vemos que existen otras tres posibilidades *abstract* (resumen), *date* (fecha del artículo) y *revhistory* (historia de revisiones). Y si miramos dentro de *<article>*, veremos que este elemento tiene dos atributos opcionales: *xreflabel*

3 Página de La Espiral.

y *lang*; conviene seleccionar el atributo *lang*, y escoger su valor *es* si el artículo va a ser escrito en español.

Algo muy importante que se debe tener en cuenta es lo siguiente: para escribir el contenido de la primera sección lo más natural era pensar que deberíamos comenzar a escribir después de su título, pero cuando usamos *Next data field* en el título, somos desplazados hasta el final del artículo. Lo que pasa, es que antes de comenzar a escribir el contenido de la sección nos falta escoger otro elemento después del título; podemos ver que la lista de posibles elementos en ese punto es la siguiente:

- *Para*. Indica un párrafo con texto. Cada párrafo en un artículo debe estar siempre entre las etiquetas *<para>* y *</para>*.
- *sect2*. Una sección de segundo nivel, o subsección.

Debemos hacer notar que los números después de *sect* no se refieren al número que identifica la sección, sino a su nivel. Existen 5 niveles de secciones:

‘Con un DTD bien diseñado, las etiquetas podrán ser introducidas fácilmente’



- **itemizedlist.** Una lista con *ítems*.
- **Orderedlist.** Una lista enumerada.
- **Variablelist.** Una lista con descripciones, ¡como la que estoy usando en este preciso momento en este artículo!
- **Figure.** Una figura.
- **Programlisting.** Un listado de un programa o unos pocos comandos, como el que se usó al comienzo de esta sección.

Dentro de un párrafo, y en general, en cualquier punto donde se pueda escribir texto, existen otros elementos útiles, como por ejemplo `<ulink>` para introducir enlaces a localizadores *url*, `<filename>` para indicar el nombre de un fichero, y `<command>` para el nombre de un comando. Si el autor usa estos elementos, después será fácil que un programa busque el nombre de los comandos o ficheros referidos en su artículo con ventajas obvias para búsquedas de información y creación de bases de datos.

TRANSFORMANDO XML EN OTROS

Antes de transformar un fichero *xml*, conviene asegurarnos de que no tenga ningún error de sintaxis. Dentro de *emacs*, se usa el comando

Validate en el menú *SGML* (o simplemente *C-c C-v*) y se pulsa *enter*.

El paquete *le-docxml* viene con unas páginas de estilo (*XSL*) que se usan por defecto para producir *html* o *latex*. Para obtener código *html* a partir de un fichero *xml* que use *LE-document.dtd*, se usa el comando *le2html*:

```
le2html fichero.xml > fichero.html
```

Si quiere usar sus propias páginas de estilo, deberá dar la opción *-s* al comando. Por ejemplo, el paquete *le-docxml* también trae las páginas de estilo que usamos en el proyecto *La Espiral*. Para producir un fichero *html* usando este estilo se debe dar el comando:

```
le2html -s laespiral.xsl fichero.xml > fichero.html
```

En *Debian*, las páginas han sido instaladas en */usr/lib/sgml/stylesheet/laespiral*, desde donde podrá copiarlas para modificarlas y crear su propio estilo. Para producir *latex*, el proceso es semejante:

```
le2latex fichero.xml > fichero.tex
```

o usando el estilo de propio de *La Espiral*.

```
le2latex -s laespiral-latex.xsl fichero.xml > fichero.tex
```



[Página principal](#) [Sobre](#) [Noticias](#) [Distribución](#) [Soporte](#) [Desarrollador](#) [Búsqueda](#)

Paquetes

¡Ojo! Esta traducción está muy desactualizada, por favor, consulte el [documento original](#).

Vea los paquetes en la distribución estable

Esta es la liberación oficial más reciente de la distribución Debian GNU/Linux, también llamada "*Main*" (principal). Todos los paquetes en ella son libres en el sentido detallado en la [Guía de Debian sobre qué es software libre](#), lo que asegura la libertad de uso y redistribución, y la disponibilidad de todo su código fuente.

También hay algunos añadidos a la distribución *main* que sufren alguna restricción que no les permite ser parte de una distribución 100% libre con Debian GNU/Linux. Pero aún así, Debian los incluye como un servicio a nuestros usuarios. Entre ellos están:

Contrib

Los paquetes de este área son libres, pero dependen de programas o hardware que no es libre.

Non-Free (no libre)

Los paquetes de este área no cuestan necesariamente dinero, pero tienen condiciones que restringen de alguna forma su uso o redistribución.

Non-US

Estos paquetes no pueden ser exportados de EEUU, y son sobre todo paquetes que incluyen software de cifrado, o software que está afectado por asuntos relacionados con patentes. La mayor parte de ellos son software libre, pero algunos no.