Tutorial de DocBook

Un enfoque integrado y a través de ejemplos

Jaime Irving Dávila

Tutorial de DocBook: Un enfoque integrado y a través de ejemplos

por Jaime Irving Dávila

Historial de revisiones

Revisión 1.0	18-06-2001	Revisado por: jid
Revisión 1.1	23-07-2001	Revisado por: jid
Revisión 1.2	24-09-2001	Revisado por: jid
Revisión 1.3	25-03-2002	Revisado por: jid
Revisión 2.0	14-04-2002	Revisado por: jid
Revisión 2.1	16-04-2002	Revisado por: jid
Revisión 2.2	2-05-2002	Revisado por: jid

Tabla de contenidos

Capítulo 1. Introducción

DocBook es un dialecto de SGML que permite la escritura de documentación técnica y que día a día gana más adeptos dentro de la comunidad del software libre y de fuente abierta. Prueba de ello es que la documentación de Gnome (`http://developer.gnome.org/projects/gdp/handbook/gdp-handbook/`), KDE (`http://i18n.kde.org/doc/markup.html`), FreeBSD (`http://www.freebsd.org/doc/en_US.ISO8859-1/books/fdp-primer/index.html`) y LinuxDoc (`http://linuxdoc.org/LDP/LDP-Author-Guide/`) esté realizada con dicho formato. De igual forma existen empresas que usan dicho para formato para generar su documentación, cómo son los caso de Cogent (`http://developers.cogentrts.com/cogent/prepdoc/book1.html`) y Mitel NSSG (`http://www.e-smith.org/docs/docprocess.html`).

Este tutorial se hizo con el objetivo de brindar un enfoque integrado para la realización de documentos en DocBook (`http://docbook.org`), usando como editor a emacs (`http://www.emacs.org`)[1] y utilizando algunas herramientas libres, así como para brindar soporte en aspectos menos documentados de DocBook como la generación de índices y la automatización de ciertas labores usando *Makefiles*; todo ello en el hermoso idioma castellano.

Se trata entonces de presentar algunos ejemplos de situaciones típicas del uso de DocBook, pero no se trata de cubrir todos los temas que están involucrados alrededor de cada tema. Por lo tanto es útil que usted tenga una guía de referencia rápida (`http://docbook.org/tdg/html/quickref.html`), en caso de que necesite profundizar en algún tema en particular

A continuación listamos algunos sitios en internet con muy buena información acerca DocBook

- El DocBook HOWTO (`htttp://wwww.ibiblio.org/godoy`) de Jorge Godoy, que posteriormente sirvió de insumo para LDP Author Guide (`http://www.tldp.org/LDP/LDP-Author-Guide`).
- Get Going with Docbook: Notes for Hackers (`http://nis-www.lanl.gov/~rosalia/mydocs/docbook-intro/docbook-intro.html`) de Mark Galassi, quizás es el primer tutorial de DocBook, fue posteriormente completado en Writing Documentation Using DocBook: A Crash Course (`http://www.bureau-cornavin.com/opensource/crash-course/index.html`).
- Self DocBook (`http://cyberelk.net/tim/docbook/selfdocbook/selfdocbook.html`) de Tim Waugh, un tutorial que es autoreferente, lleno de trucos interesantes y no tan documentados de DocBook
- NewbieDoc Docbook Guide (`http://newbiedoc.sourceforge.net/metadoc/docbook-guide.en.html`), de Jesse Goerz el cuál es una introducción muy completa para usuarios novatos y Mark Johnson Bookmarks (`http://www.dulug.duke.edu/~mark/bookmarks/docbookmarks.html`), que es una página llena de enlaces muy buenos referentes a DocBook.
- El Tutorial de DocBook (`http://lucas.hispalinux.es/Presentaciones/0000otras/conf-olea1/html`) de Ismael Olea, que muestra la forma en que funciona un ambiente DocBook, así como los objetivos que persigue dicho formato y los Recursos básicos para empezar con docbook (`http://www.olea.org/docbook/index.html`), que son una colección de enlaces acerca de DocBook. Una de las características más importantes de dicha documentación es que esta disponible en el hermoso idioma castellano.

Sin embargo la guía definitiva para DocBook sigue siendo el libro DocBook: The Definitive Guide (`http://docbook.org/tdg/en/html/docbook.html`). En la fecha de revisión de este documento la última versión era la 2.0.4 y era documentación libre publicada bajo la GFDL. De la versión anterior

(1.0.3) usted puede obtener una copia (`http://www.oasis-open.org/docbook/documentation/reference/html.zip`) de dicho libro para uso personal.

Agradezco las valiosas sugerencias de Vladimir Támara <`vladimir@tamarapatino.com`> y Alejandro Forero <`azul@bachue.com`> quienes señalaron diversos errores e imprecisiones en este documento. Para cualquier sugerencia o duda que contribuya a hacer de este un mejor tutorial no dude en enviarme un email a <`jadavila@uniandes.edu.co`> Las fuentes completas de este libro así de sus ejemplos se encuentran en este archivo (`dbktut.tar.gz`), siéntase libre de usar éstas como mejor le convenga.

Sólo me resta desearle que se divierta tanto aprendiendo, como yo disfruté escribiendo este documento.

Notas

1. Para aquellos que usan como editor a vim (`http://vim.org`), listo un par de enlaces que les serán de utilidad para configurar su editor de manera óptima. El primero es el archivo de configuración (`http://www.lodestar2.com/software/docbook/vimrc`) de Dan York, el segundo un *script* basado en el anterior, para obtener un modo Docbook para vim (`http://vim.sourceforge.net/scripts/script.php?script_id=38`)

Capítulo 2. Escribiendo un ejemplo básico

En este capítulo introduciremos la manera de crear un pequeño documento en DocBook, usando emacs. Trataremos de ser didácticos, por lo cuál existirán a veces demasiadas instrucciones sobre todo referentes a la interacción con emacs. El lector más aventajado podrá omitir tales instrucciones con toda tranquilidad.

Algunas instrucciones preliminares

1. A continuación listamos las herramientas que usamos así como sus últimas versiones a la fecha de la escritura de este documento:

Aplicación	DTD de DocBook	*Versión*	4.1
Enlace	`http://www.oasis-open.org/committees/docbook/sgml/4.1/index.shtml`		
Aplicación	DSSL de DocBook	*Versión*	1.76
Enlace	`http://docbook.sourceforge.net/projects/dsssl/`		
Aplicación	OpenJade	*Versión*	1.3
Enlace	`http://openjade.sourceforge.net/`		
Aplicación	Jadetex	*Versión*	4.1
Enlace	`http://sourceforge.net/projects/jadetex/`		
Aplicación	docbook-utils	*Versión*	0.6.10
Enlace	`ftp://sources.redhat.com/pub/docbook-tools/new-trials/SOURCES/docbook-utils-0.6.10.tar.gz`		
Aplicación	emacs	*Versión*	21.2
Enlace	`http://www.gnu.org/software/emacs/emacs.html`		
Aplicación	PSGML	*Versión*	1.2.5
Enlace	`http://www.lysator.liu.se/~lenst/about_psgml/` y `http://psgml.sourceforge.net/`		
Aplicación	make	*Versión*	3.78.1
Enlace	`http://www.gnu.org/software/make/make.html`		

Importante: Es usual que cualquier distribución de Linux traiga dichas herramientas instaladas, en caso contrario puede encontrar información general de como instalarlas en Linux en el DocBook Install mini-HOWTO (`http://www.linuxdoc.org/HOWTO/mini/DocBook-Install/`) o en DocBook XML Install HOWTO (`http://www.jimweller.net/jim/dbxmlqs/index.html`) si bien hay también información específica en el caso de las distribuciones Debian (`http:/people.debian.org/~bortz/SGML-HOWTO/potato/howto.html`) y RedHat (`http://sources.redhat.com/docbook-tools`). De igual forma hay paquetes integrados en los cuales ya vienen todas las herramientas necesarias como lo son los del TEI (`http://www.tei-c.org/Software/tei-emacs.tar.gz`) y XAE (`http://xae.sunsite.dk`)

Sugerencia: Cabe aclarar que dichas herramientas se encuentran también disponibles para Windows. Se pueden encontrar instrucciones detalladas de como instalarlo en la página de SGML for Windows NT (`http://ourworld.compuserve.com/homepages/hoenicka_markus/ntsgml.html`) (en inglés) o en la página de Camilo Camacho (`http://abadon.uniandes.edu.`

`co/ingsw/documentos/tutorial/windows.html\#AEN853`) (en castellano). De igual forma hay paquetes como los del TEI (`http://www.tei-c.org/Software/tei-emacs.exe`) y XAE (`http://xae.sunsite.dk`), en los cuales vienen preinstaladas dichas utilidades.

2. Para que tenga facilidades adicionales de edición, a las que tiene el modo PSGML de emacs, es necesario modificar el archivo de configuración `.emacs`. Una manera de hacer esto es bajar el archivo `dotemacs` y usar un comando como el siguiente:

   `[irving@abadon e1]$` **`cat dotemacs »~/.emacs`**

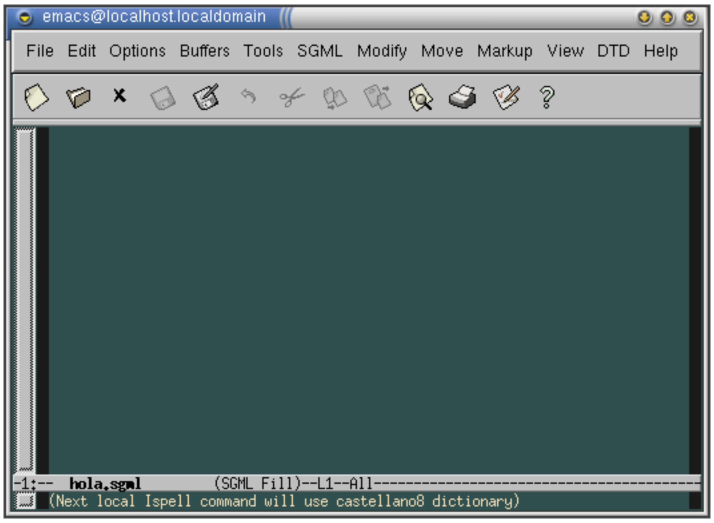
   Dicho comando simplemente añade algunas líneas al archivo `~/.emacs`, creándolo si no existe. El archivo `dotemacs` de configuración está basado en la página PSGML tricks (`http://www.snee.com/bob/sgmlfree/emcspsgm.html`), en dicho sitio podrá encontrar más información al respecto

Haciendo un documento pequeño

Para comenzar a escribir un documento necesitará abrir una sesión de emacs, para ello desde la línea de comandos del shell, ejecute **`emacs hola.sgml&`**, en mi computador esto luce de la siguiente forma:

`[irving@abadon e1]$` **`emacs hola.sgml &`**

Una vez hecho esto obtendrá una pantalla como la siguiente, observe que aparecen algunos menus como SGML, DTD, Markup y otros. En caso de no ser así, revise la instalación de su sistema.



Una vez en el editor, debe comenzar escribiendo la línea que describe que el documento SGML que usted escribirá es de tipo DocBook, para ello solo escriba las líneas **`<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.1//EN">`**[1], una manera más automática de hacer esto, aprovechando la configuración del archivo `.emacs` es mediante los menus DTD⟶Insert DTD⟶DocBook 4.1.

En emacs es usual que cualquier conjunto de comandos que usted pueda realizar mediante menus los pueda realizar también usando un conjunto de comandos desde el teclado, esto permite que cuando un comando es usado muy frecuentemente se pueda realizar de forma más rápida usando un atajo (*shortcut*) apropiado. En este caso el atajo correspondiente será C-c-C-u-C-d

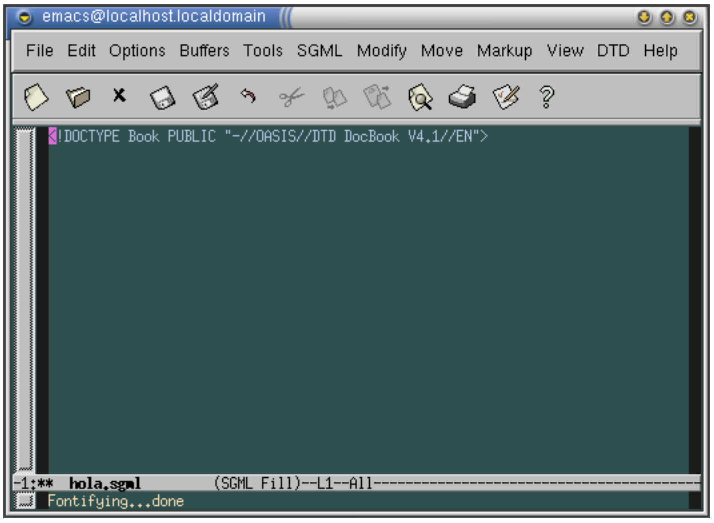
> **Importante:** Cuando escribamos por ejemplo C-c queremos decir que presione **Control** y al mismo tiempo **C**. Cuando escribamos por ejemplo M-x, queremos decir que presione la tecla **Alt** y al mismo tiempo la tecla x. En algunas ocasiones escribiremos M-C-x, lo cual quiere decir presionar al tiempo las teclas **Control**,**Alt** y **X**.

Después de esta secuencia de teclas es necesario que escriba **`DocBook 4.1`** y luego **ENTER**. Cuando nos encontremos en situaciones como éstas, en las que una misma acción se puede realizar mediante una escogencia en el menú y un atajo en el teclado lo escribiremos usando una abreviación como la siguiente DTD⟶Insert DTD (**C-c-C-u-C-d**)

> **Sugerencia:** Para evitar equivocarse escribiendo las palabras **`Docbook 4.1`** usted puede usar una característica muy útil de emacs que es la autocompletación. Esta consiste en que haciendo uso de la tecla **TAB** se pueden ir presentando las opciones válidas hasta el momento. Es así como si en el anterior caso hubiera digitado **TAB** después de haber hecho C-c-C-u-C-d se le hubieran presentado las opciones posibles que eran `HTML 4` y `DocBook 4.1` escribiendo **D TAB** la única opción posible será entonces **DocBook 4.1**, restando solo escribir **Enter**

A continuación es necesario cargar la sintaxis de DocBook, para que el editor muestre en diferente fuente aquellas partes del documento que son marquillas (o *tags*). Esto lo puede hacer mediante DTD⟶Parse DTD (**C-c C-p**), una vez hecho esto emacs comenzará a cargar la información de sintaxis de DocBook y posteriormente se obtendrá un color distinto (usualmente púrpura) en el encabezado que usted acabó de imprimir como lo muestra la siguiente pantalla

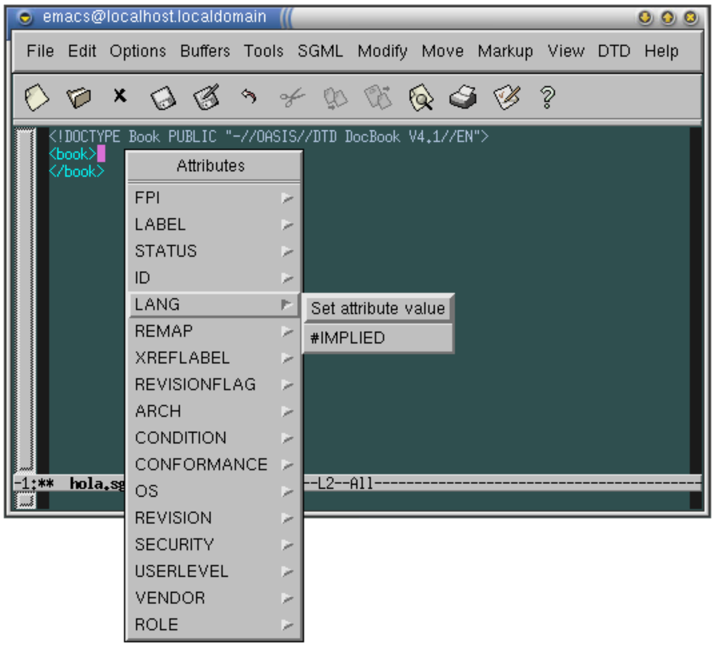
> **Sugerencia:** En caso de que emacs presente un error significa que la versión 4.1 no se encuentra correctamente instalada, use un número de versión más baja (3.1 por ejemplo) o instale la última versión.



En una línea aparte es necesario colocar la primer marquilla del documento que es la que corresponde a libro (`book`), la primer forma consiste en escribir **`<book>`**, usando el teclado. Una forma mucho más eficiente de introducir marquillas es usando C-c-C-e , a continuación aparecerá un mensaje en el buffer de emacs con el mensaje `Element: book`, indicando que la única marquilla disponible es ésta, posteriormente pulse **Enter**, creándose las marquillas de inicio y cerrado de un libro (`book` y `/book` respectivamente)

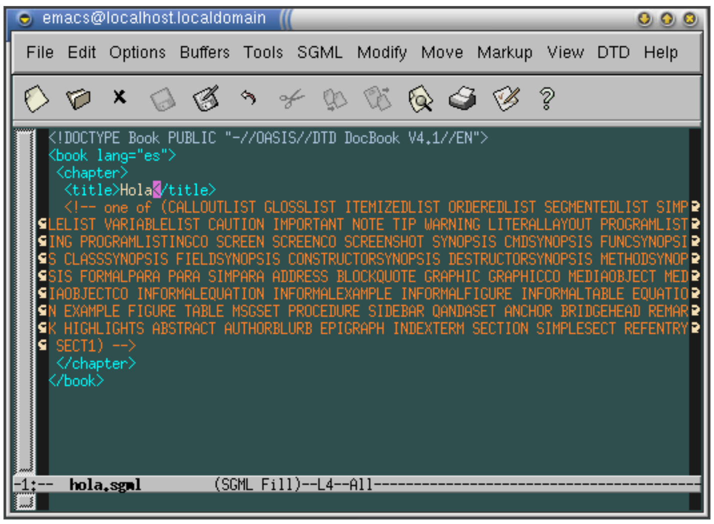
Una forma alternativa de elegir que marquilla insertar se puede hacer mediante Markup⟶Insert Element (**Shift+Botón Derecho**), y luego señalar la marquilla a utilizar, que en este caso es `book`

Como nuestro primer ejemplo estará escrito en castellano, es necesario indicar el idioma a que vamos a usar, para ello necesitamos editar uno de los atributos de la marquilla <book>; através de menúes esto se puede hacer con Markup⟶Insert Attribute (**Botón Derecho**). A continuación se presentarán los diferentes atributos de dicha marquilla, es necesario entonces seleccionar LANG⟶Set Attribute, a continuación en el buffer aparecerá `Value for LANG(CDATA):` , por lo que habrá que escribir **`es`**. Tendremos entonces que nuestro editor lucirá de la siguiente forma



Si uno sólo quiere utilizar el teclado para introducir dicho atributo puede usar C-c **+**, con lo que el editor le responderá `Attribute name` (que en este caso es `lang`) y luego le preguntará `Value for LANG in BOOK (CDATA)` (que en este caso es `es`). Recuerde que cuando se le solicita el nombre del atributo, usted puede usar autocompletación, esto hará que su labor sea más efectiva y menos susceptible de errores

Ahora introduciremos un capítulo, usando C-c C-e seguido de `Chapter`, una vez hecho esto se escribirá el título del capítulo que será **`Hola`**, tenemos entonces que el editor luce de la siguiente forma:



Los comentarios en DocBook se comienzan y terminan usando los símbolos <!– y –>, es de notar entonces que lo que aparece en letra roja es un comentario insertado automáticamente por el modo PSGML que indica cuáles marquillas están disponibles a continuación. Una manera alternativa de conocer cuáles son las marquillas disponibles es usar C-c C-t. Si desea borrar los comentarios recién insertados, muévase al inicio de la línea donde está ubicado el comentario (Aquí puede ser útil C-a, que mueve al inicio de una línea) y luego pulse C-k (que se encarga de borrar la línea)

> **Sugerencia:** En caso de que desee desactivar la característica de inserción automática de comentarios con marquillas disponibles , esto se puede hacer con SGML+User Options⟶Insert Missing Element o incluyendo la siguiente línea en el archivo `.emacs`, **`(setq sgml-insert-missing-element-comment nil)`**.

Lo que haremos a continuación es escribir un párrafo con la introducción del capítulo. Una aclaración importante es que todo párrafo en DocBook debe ubicarse dentro de las marquillas `para` y `/para`. Dicha marquilla puede introducirse en emacs cómo lo hemos mostrado en el caso de la marquilla `book` o `chapter` , pero debido a que es de tan frecuente uso, la introdujimos en nuestro archivo `.emacs` y solo basta usar C-c **p**. Una vez hecho esto introduciremos el párrafo **`Esto es la introducción`**

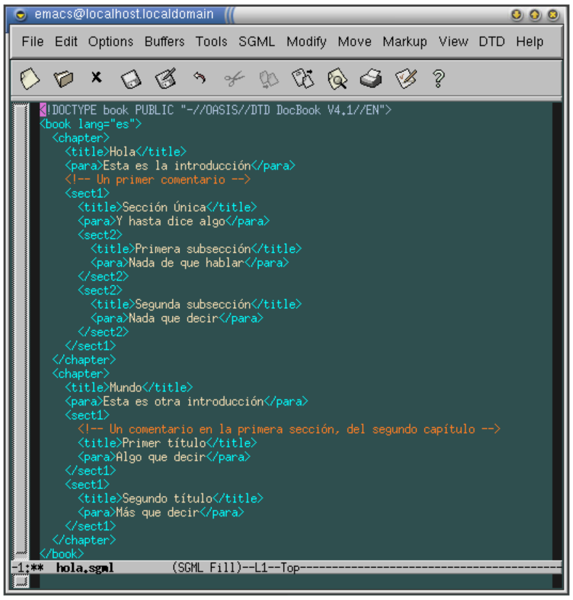
Nuestra idea es crear un pequeño libro con dos capítulos y varias secciones, subsecciones, párrafos entre ellos; para introducir una sección dentro de un capítulo la marquilla de utilidad será `sect1`, en caso de querer hacer una subsección dentro de ésta basta usar `sect2` y así sucesivamente.

Para introducir comentarios, la configuración local nos permite hacerlo a través de C-c **o**. Con esto en mente vemos que és facil escribir el siguiente código final de nuestro primer ejemplo

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
<book lang="es">
  <chapter>
    <title>Hola</title>
    <para>Esta es la introducción</para>
    <!- Un primer comentario ->
    <sect1>
      <title>Sección Única</title>
      <para>Y hasta dice algo</para>
      <sect2>
<title>Primera subsección</title>
<para>Nada de que hablar</para>
      </sect2>
      <sect2>
<title>Segunda subsección</title>
<para>Nada que decir</para>
      </sect2>
    </sect1>
  </chapter>
  <chapter>
    <title>Mundo</title>
    <para>Esta es otra introducción</para>
    <sect1>
      <!- Un comentario en la primera sección, del segundo capítulo ->
      <title>Primer título</title>
      <para>Algo que decir</para>
    </sect1>
    <sect1>
      <title>Segundo título</title>
      <para>Más que decir</para>
```

```
    </sect1>
  </chapter>
</book>
```

o gráficamente,



De vez en cuando es útil saber dentro de que marquilla se encuentra el cursor ubicado, para ello basta usar SGML⟶List Valid Tags (**C-c C-t**). Por último en caso de que necesite saber los diversos comandos que usted puede usar en el modo PSGML de emacs basta que use C-c C-h, o visite una Guía de Referencia Rápida de EMACS PSGML (`http://www.snee.com/bob/sgmlfree/psgmqref.html`)

Visualización del documento

Validando el SGML

Antes de generar cualquier tipo de documento[2] a partir de la fuente SGML, es necesario que se trate de un documento válido, es decir que se respete la sintaxis de DocBook. Para ello basta usar el comando C-c C-v **Enter**, en caso de que no haya salvado aparecerá un mensaje en el buffer pidiéndole que lo haga, para lo cual debe escribir **`y`**.

A continuación se dividirá la ventana y aparecerá un mensaje que indica el resultado de la validación exitosa. Más explícitamente dicho mensaje es:

```
cd /home/irving/20013/doc-tut/e1
nsgmls -s hola.sgml
SGML validation finished at Sat Jun 16 07:38:03
```

> **Sugerencia:** Una vez leído este mensaje es muy probable que desee salirse del modo de dos ventanas. Para ello teniendo el cursor en la ventana superior, use C-x **1**

Generación de Documentos

Existen muchas herramientas para convertir archivos de DocBook a diversos formatos. Muy probablemente dentro de su sistema Linux usted contará con las siguientes:

db2html o **docbook2html**

Permite la conversión de formato DocBook a formato HTML

db2ps o **docbook2ps**

Permite la conversión de formato DocBook a formato ps

db2pdf o **docbook2pdf**

Permite la conversión de formato DocBook a formato pdf

db2rtf o **docbook2rtf**

Permite la conversión de formato DocBook a formato rtf

El uso de tales herramientas es bastante sencillo. Por ejemplo si uno desea convertir el documento al formato HTML, basta escribir, desde la interfaz de comandos, **`db2html`** *`nomarch`*. Dentro de emacs se puede hacer un llamado a la línea de comandos usando M-!. Basta usar M-! y luego digitar **`db2html`** `hola.sgml`, para convertir el documento a HTML sin tener que salir de emacs. A continuación se dividirá la pantalla y en la segunda ventana se obtendrá un resultado similar al siguiente:

```
TMPDIR is DBTOHTML_OUTPUT_DIR4724
 input file was called hola.sgml - output will be in hola
working on ../hola.sgml
about to copy cascading stylesheet and admon graphics to temp dir
about to rename temporary directory to hola
```

Atención

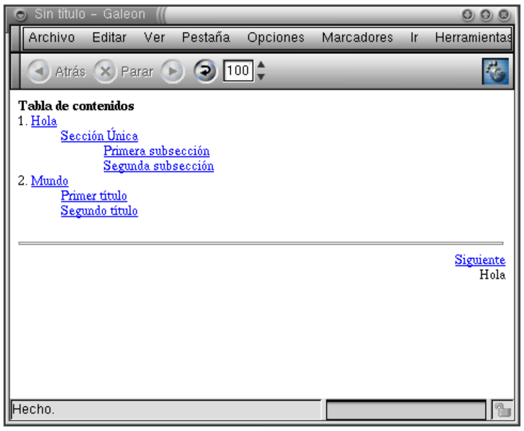
Cabe anotar que en la distribución Debian no se tienen herramientas con estos nombres. En vez de éstas se usa sgmltools con opciones apropiadas. Por ejemplo en vez de escribir **`db2html`** `hola.sgml`, se usa **`sgmltools -b html`**`hola.sgml` o en vez de **`db2ps`** `hola.sgml` se usa **`sgmltools -b ps`** `hola.sgml`. El paquete sgmltools se encuentra disponible en `http://sgmltools-lite.sourceforge.net`

Sugerencia: db2html es simplemente un *script* llama a una aplicación como openjade (o jade), para procesar un archivo sgml usando unas determinadas hojas de estilo. En caso de no querer usar dicho *script* basta con emplear un comando como el siguiente

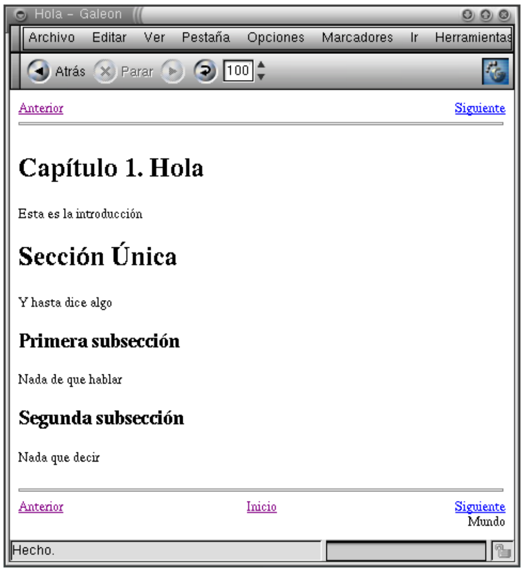
```
openjade -t sgml -ihtml -d/usr/share/sgml/docbook/dsssl/nwalsh-modular/html/docbook.dsl hola.
sgml
```

> El archivo especificado después de la opción `-d` es la dsssl usada por DocBook, para averiguar donde se encuentra dicho archivo puede usar el comando **`find / -name docbook.dsl`**

Dicha aplicación creará un directorio `hola`, en donde almacenará el conjunto de páginas que conforman su documento. La página principal contendrá la tabla de contenidos de su documento y se llamará `book1.html`. En nuestro caso dicha página luce de la siguiente forma



De igual forma, si hacemos click en el enlace de Siguiente, obtenemos que la página del primer capítulo luce de la siguiente forma:



En caso de que se quiera generar el documento en formato PS basta usar **`ESC !`** y luego digitar **`db2ps hola.sgml`**

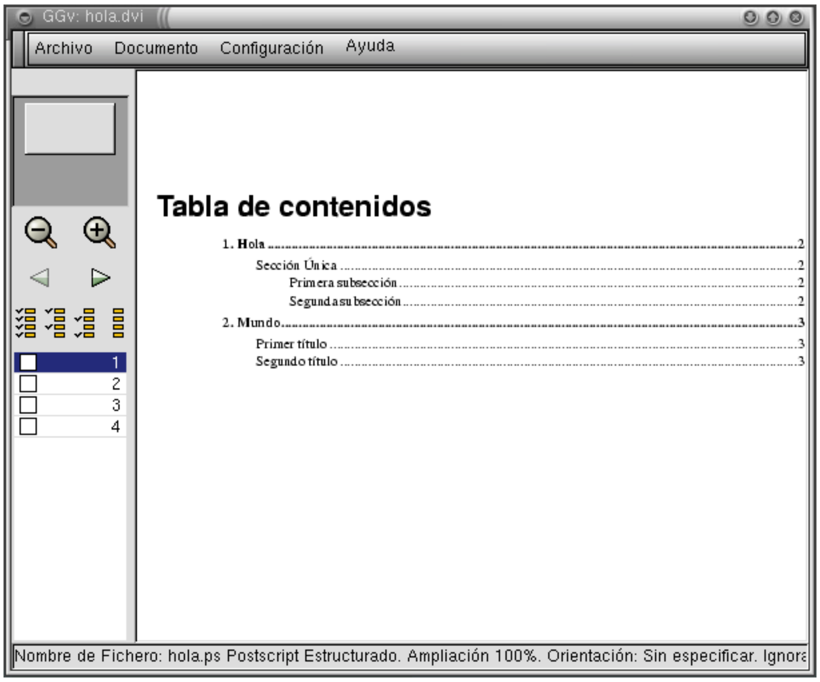
En la mitad inferior de emacs se obtendrá un conjunto de mensajes que indican que se está generando el documento. Las últimas líneas de este mensaje lucen de la siguiente forma:

```
Output written on hola.dvi (4 pages, 26792 bytes).
Transcript written on hola.log.
This is dvips(k) 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
' TeX output 2001.06.16:0857' -> hola.ps
<texc.pro><8r.enc><texps.pro><special.pro><color.pro>. [1] [2] [3] [4]
```

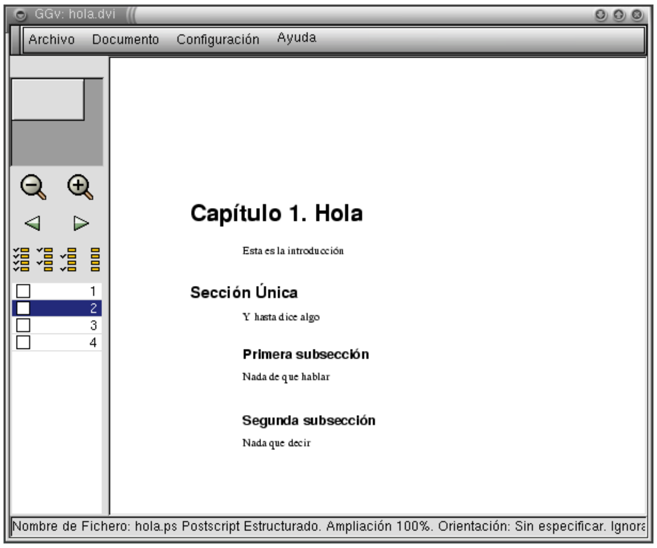
Atención

En algunos casos cuando se procesa este documento se presentan mensajes de error como `l.42 \select@language{spanish}` . Para corregir este tipo de errores es aconsejable tener la última versión de jadetex, como se recomienda en `http://lists.oasis-open.org/archives/docbook-apps/200005/msg00131.html`. En caso de mayores problemas puede consultar `http://www.olea.org/docbook/problemas-db-es.html`

Esto generará el archivo `hola.ps`, el cuál puede ser visualizado usando GGv. Para verlo use **`M-!`** y digite **`ggv hola.ps &`**. Obtendrá una pantalla como la siguiente



Avanzando de página, veremo el primer capítulo, que luce de la siguiente forma



El proceso de generar PDF y RTF es muy similar a lo visto hasta el momento, por lo que no consideramos importante describirlo detalladamente.

Notas

1. En la fecha en que este documento escrito, la versión estable de DocBook era la 4.1, por ende esta será nuestra referencia
2. Hay que recordar que se pueden generar documentos en formato HTML, PS, PDF y RTF

Capítulo 3. Escribiendo un documento más complejo

En el capítulo anterior introdujimos sólo los elementos más básicos de DocBook y la forma de editar dichos documentos usando emacs. En este capítulo introduciremos características más interesantes, que nos permitirán escribir documentos más complejos. Nuestra aproximación será ir introduciendo progresivamente dichas características a medida que se necesiten dentro de un pequeño documento llamado `herramientas.sgml` que se muestra en la la sección de nombre *Listado de `herramientas.sgml`*.

Un esbozo de documento

Con el conocimiento que tenemos es fácil escribir el documento que presentamos a continuación

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
<book lang="es">
  <chapter>
    <title><acronym>DocBook</acronym></title>
    <para></para>
    <sect1>
      <title>Historia</title>
      <para></para>
    </sect1>
    <sect1>
      <title>Marquillas</title>
      <para></para>
    </sect1>
  </chapter>
  <chapter>
    <title><application>emacs</application></title>
    <para></para>
    <sect1>
      <title>Invocación</title>
      <para></para>
    </sect1>
    <sect1>
      <title>Escribiendo y guardando un archivo</title>
    </sect1>
  </chapter>
</book>
```

En dicho ejemplo introdujimos la estructura de un documento que queremos escribir, los únicos elementos nuevos que ingresamos son `acronym` y `application`, que básicamente corresponden a un acrónimo (o una sigla como lo es SGML) y a una aplicación (es decir un software de aplicación, como lo es el editor emacs)

Varios tipos de listas

Vamos a comenzar a incluir información dentro del capítulo de DocBook de nuestro documento `herramientas.sgml`. Lo primero que haremos en este sentido será incluir un pequeño párrafo.

> **Sugerencia:** Cuando un párrafo supera una línea de longitud, este se puede alinear fácilmente a las marquillas correspondientes (para facilidad de lectura) usando M-q

Además necesitaremos introducir una lista, para lo cual es necesario el uso de las marquillas `itemizedlist` y `listitem`

> **Importante:** Nótese que cuando se introduce la marquilla `itemizedlist` usando C-c C-e, inmediatamente se incluyen marquillas de `listitem`

Veamos a continuación dicho ejemplo y como es visualizado.

```
<sect1>
  <title>Historia</title>
  <para><acronym>DocBook</acronym> es un lenguaje de marcado, que
  permite escribir documentación técnica, nacido en 1991.</para>

  <para>Los principales contribuyentes a dicho proyecto han
  sido:</para>
  <itemizedlist>
    <listitem>
      <para>Hal Computer Systems y O'Reilly &amp; Associates, de
        1991 a 1994</para>
    </listitem>
    <listitem>
      <para>El grupo Davenport, de 1994 a 1998.</para>
    </listitem>
    <listitem>
      <para>El grupo <acronym>OASIS</acronym> de 1998 hasta hoy.</para>
    </listitem>
  </itemizedlist>
```

Y se el contenido se ve como:

DocBook es un lenguaje de marcado, que permite escribir documentación técnica, nacido en 1991.

Los principales contribuyentes a dicho proyecto han sido:

- Hal Computer Systems y O'Reilly & Associates, de 1991 a 1994
- El grupo Davenport, de 1994 a 1998.
- El grupo OASIS de 1998 hasta hoy.

En caso de que uno desee una lista numerada, la marquilla de utilidad será `orderedlist`, el resultado se muestra a continuación.

1. Hal Computer Systems y O'Reilly & Associates, de 1991 a 1994
2. El grupo Davenport, de 1994 a 1998.
3. El grupo OASIS de 1998 hasta hoy.

> **Sugerencia:** Para cambiar la marquilla `itemizedlist` por `orderedlist` basta que se ubique dentro de la marquilla `itemizedlist` y haga Modify⟶Change Element Name (**C-c =**), luego escriba **`orderedlist`** (recuerde que puede usar autocompletación, así que digitando **`or`** y **`TAB`** bastará)

El estilo de numeración de tales listas es el atributo **`numeration`** de la marquilla `orderedlist`, en caso de cambiar ésta por **`<orderedlist numeration="lowerroman">`** , el resultado sería:

i. Hal Computer Systems y O'Reilly & Associates, de 1991 a 1994
ii. El grupo Davenport, de 1994 a 1998.
iii. El grupo OASIS de 1998 hasta hoy.

> **Sugerencia:** Una manera adicional de asignar atributos a una marquilla, es ubicándose en ella y usar Modify⟶Edit Atributes... (**C-c C-a**), luego se edita el atributo que se desee moviéndose con **TAB**, borrándolo (C-c C-a) y luego salvándolo (C-c C-c)

Tablas y caracteres especiales

A continuación mostramos el código de la primera sección de nuestro documento de ejemplo y la manera cómo se le da formato a dicha información. Posteriormente explicaremos algunas de las marquillas y características que este código usa.

```
<sect1>
  <title>Marquillas</title>
    <para>En <acronym>DocBook</acronym>, la estructura de un
    documento se se marca a través de marquillas de inicio y
    fin. Dichas marquillas lucen correspondientemente como
    <replaceable>&amp;lt;marquilla></replaceable> y
    <replaceable>&amp;lt;\marquilla></replaceable>.</para>

    <para>A continuación mostramos una tabla con algunas de las
    marquillas más usadas</para>
    <table>
```

```
      <title>Algunas marquillas</title>
      <tgroup cols="2">
<thead>
  <row>
    <entry>Nombre de la marquilla</entry>
    <entry>Descripción de la marquilla</entry>
  </row>
</thead>
        <tbody>
          <row>
            <entry>Nombre de la marquilla</entry>
            <entry>Descripción de la marquilla</entry>
          </row>
          <row>
            <entry>&lt;book>/entry>
            <entry>Es la más importante, indica el inicio y fin de
            un libro</entry>
          </row>
          <row>
            <entry>&lt;chapter></entry>
            <entry>Indica el inicio y fin de un capítulo</entry>
         </row>
       </tbody>
    </tgroup>
  </table>
</sect1>
```

En DocBook, la estructura de un documento se se marca a través de marquillas de inicio y fin. Dichas marquillas lucen correspondientemente como <*marquilla*> y <*\marquilla*>.

A continuación mostramos una tabla con algunas de las marquillas más usadas

Nombre de la marquilla	**Descripción de la marquilla**
<book>	Es la más importante, indica el inicio y fin de un libro
<chapter>	Indica el inicio y fin de un capítulo

Hay algunos caracteres reservados para DocBook, como lo son < y & , por eso en caso de que uno desee escribir alguno de éstos debe reemplazarlos por < y &, respectivamente. Las secuencias de texto que, como en el anterior caso, comienzan por & y terminan en ; se llaman *entidades* en la terminología SGML/XML

> **Sugerencia:** En DocBook existen una gran cantidad de *entidades* para describir todo tipo de caracteres como lo son por ejemplo o distintos tipos de flechas. Desde emacs usted puede encontrar tal tipo de entidades a través de Markup⟶Insert Entity

Sugerencia: En caso de necesitar teclear <, puede usar C-c <

De otro lado para hacer una tabla es necesario primero declararla usando `table` y luego se indica mediante la marquilla `tgroup` y su atributo **`cols`** el número de columnas de tal tabla. Cada fila se indica usando `row` y para hacer referencia a una celda dentro de una fila se usa `entry`. Si se usa el modo PSGML de emacs, la escritura de muchas de éstas marquillas es generada automáticamente.

Mostrando pantallas y capturas de pantallas

Cuando se escribe documentación técnica es muy útil mostrar pantallas que permitan visualizar el conjunto de acciones que se está realizando. DocBook permite hacer esto mediante la marquilla `screen`, a continuación mostramos un ejemplo de este tipo de codificación y cuál es su apariencia en el documento final

```
<chapter>
  <title><application>emacs</application></title>
  <sect1>
    <title>Invocación</title>
    <para>Para invocar a <application>emacs</application>, basta
    hacer lo siguiente desde una línea de comandos</para>

<screen><prompt>[irving@abadon e2]$ </prompt><userinput>emacs&</userinput>
<computeroutput>[1] 6251</computeroutput>
<prompt>[irving@abadon e2]$ </prompt>
</screen>
```

Para invocar a emacs, basta hacer lo siguiente desde una línea de comandos

```
[irving@abadon e2]$ emacs&
[1] 6251
[irving@abadon e2]$
```

La marquilla `screen` sirve para indicar que se el texto escrito en ella se trata de una captura de pantalla, es de notar que el texto dentro de ella es copiado tal cual, por lo cual importan los espacios y los cambios de línea.

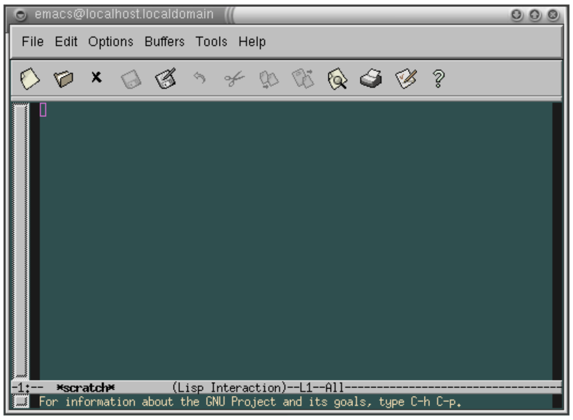
Las marquillas `prompt`, `computeroutput` y `userinput`, sirven respectivamente para denotar el *prompt* (en una sesión de línea de comandos de UNIX), un texto que es generado por el computador y una entrada del usuario respectivamente

Con frecuencia, se desea capturar pantallas y luego utilizarlas dentro de un documento, el siguiente es un ejemplo de tal situación.

```
<para>Esto produce que se abra una ventana de
<application>emacs</application> como la siguiente:</para>
  <informalfigure>
    <screenshot>
      <mediaobject>
        <imageobject>
```

```
          <imagedata fileref="emacs.eps" format="eps" scale="40">
        </imageobject>
        <imageobject>
          <imagedata fileref="emacs.png" format="png">
        </imageobject>
        <textobject>
          <phrase>Una ventana de
          <application>emacs</application></phrase>
        </textobject>
      </mediaobject>
    </screenshot>
  </informalfigure>
```

Esto produce que se abra una ventana de emacs como la siguiente:



> **Sugerencia:** Para capturar pantallas, se recomienda usar Gimp (`http://gimp.org`) a través de Fichero⟶Adquirir⟶Captura de pantalla.

La marquilla `mediaobject` hace referencia a que a continuación se encuentra un objeto, que puede tener dos representaciones gráficas (para ello es la marquilla `imageobject`) y una textual (para ello es la marquilla `textobject`). Cada representación gráfica tiene un formato en particular (aquí se usa el atributo **`format`**), el formato eps es usado para la salida en ps y el formato png es usado para la publicación en la red. Notemos que adicionalmente en el caso del eps usamos reducción de escala en un 40% para que dicha gráfica quepa mejor en el documento.

> **Importante:** Es necesario que los archivos `emacs.eps` y `emacs.png` se encuentren en la misma ruta que el archivo `herramientas.sgml`, para cuando se haga la conversión a través de db2html o db2ps

Describiendo interacción con la aplicación

En el siguiente ejemplo describiremos la manera como interactúa el usuario con una aplicación a través de los menús o mediante combinaciones de teclado

```
<sect1>
  <title>Escribiendo y guardando un archivo</title>

    <para>Dentro del editor escriba la frase <userinput>Hola
    Mundo</userinput> y posteriormente guarde dicho archivo usando
    <menuchoice>
      <shortcut>
        <keycombo action="seq">
  <keysym>C-x</keysym><keysym>C-s</keysym>
</keycombo>
      </shortcut>
      <guimenu>Files</guimenu> <guimenuitem>Save Buffer
      as</guimenuitem>
    </menuchoice>, a continuación <application>emacs</application>
    responderá con el mensaje <computeroutput>File to save
    in:~/20013/doc-tut/e2</computeroutput>, restando que usted
    escriba solamente el nombre del archivo
   (<filename>hola.txt</filename>).</para>
</sect1>
```

y la manera como se ve dicho código es

Dentro del editor escriba la frase **`Hola Mundo`** y posteriormente guarde dicho archivo usando Files⟶Save Buffer as (**C-x C-s**), a continuación emacs responderá con el mensaje `File to save in:~/20013/doc-tut/e2`, restando que usted escriba solamente el nombre del archivo (`hola.txt`).

Dentro de este ejemplo hemos usado la marquilla `menuchoice` que indica la interacción con la aplicación através de menús. Para seleccionar cada interacción con el menú lo hacemos a través de `guimenu` y `guimenuitem`.

> **Sugerencia:** Existen marquillas como `guibutton` y `guiicon` que permiten describir la interacción con botones o iconos dentro de una aplicación.

Otro punto importante de este ejemplo consiste en la utilización de `shortcut` que indica un atajo de interacción a través del teclado. Dentro de este usamos `keycombo`, que nos permite describir una combinación de teclas. Tal marquilla tiene el atributo *`action`*, el cual sirve para describir como se efectúa la combinación de teclas, que puede ser secuencial (`seq`) o simultánea (`simul`). La marquilla `keysym` se usa cuando se quiere indicar un símbolo que significa una tecla (o combinación de teclas) especial.

La última observación consiste en que cuando se haga referencia a un nombre de un archivo es conveniente usar `filename`. En caso de que uno hable de un directorio debe usar el atributo de dicha marquilla *`class`* igual a `directory`

Listado de `herramientas.sgml`

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
<book lang="es">
  <chapter>

    <title><acronym>DocBook</acronym></title>
    <sect1>
      <title>Historia</title>
      <para><acronym>DocBook</acronym> es un lenguaje de marcado, que
       permite escribir documentación técnica, nacido en 1991.</para>

      <para>Los principales contribuyentes a dicho proyecto han
      sido:</para>
      <itemizedlist>
<listitem>
  <para>Hal Computer Systems y O'Reilly &amp; Associates, de
       1991 a 1994</para>
</listitem>
<listitem>
  <para>El grupo Davenport, de 1994 a 1998.</para>
</listitem>
<listitem>
  <para>El grupo <acronym>OASIS</acronym> de 1998 hasta hoy.</para>
</listitem>
      </itemizedlist>
    </sect1>
    <sect1>
      <title>Marquillas</title>

      <para>En <acronym>DocBook</acronym>, la estructura de un
      documento se se marca a través de marquillas de inicio y
      fin. Dichas marquillas lucen correspondiente como
      <replaceable>&lt;marquilla></replaceable> y
      <replaceable>&lt;\marquilla></replaceable>.</para>

      <para>A continuación mostramos una tabla con algunas de las
      marquillas más usadas</para>

      <table>
<title>Algunas marquillas</title>
<tgroup cols="2">
  <thead>
    <row>
      <entry>Nombre de la marquilla</entry>
      <entry>Descripción de la marquilla</entry>
    </row>
  </thead>
  <tbody>
    <row>
      <entry><sgmltag>book</sgmltag></entry>
      <entry>Es la más importante, indica el inicio y fin de
      un libro</entry>
```

```
    </row>
    <row>
      <entry><sgmltag>chapter</sgmltag></entry>
      <entry>Indica el inicio y fin de un capítulo</entry>
    </row>
  </tbody>
</tgroup>
      </table>
    </sect1>
  </chapter>
  <chapter>
    <title><application>emacs</application></title>
    <sect1>
      <title>Invocación</title>
      <para>Para invocar a <application>emacs</application>, basta
      hacer lo siguiente desde una línea de comandos</para>
<screen><prompt>[irving@abadon e2]$ </prompt><userinput>emacs&</userinput>
<computeroutput>[1] 6251</computeroutput>
<prompt>[irving@abadon e2]$ </prompt>
</screen>
      <para>Esto produce que se abra una ventana de
      <application>emacs</application> como la siguiente:</para>

      <informalfigure>
<screenshot>
  <mediaobject>
    <imageobject>
      <imagedata fileref="emacs.eps" format="eps" scale="40">
    </imageobject>
    <imageobject>
      <imagedata fileref="emacs.png" format="png">
    </imageobject>
    <textobject>
      <phrase>Una ventana de
      <application>emacs</application></phrase>
    </textobject>
  </mediaobject>
</screenshot>
      </informalfigure>
    </sect1>
    <sect1>
      <title>Escribiendo y guardando un archivo</title>

      <para>Dentro del editor escriba la frase <userinput>Hola
      Mundo</userinput> y posteriormente guarde dicho archivo usando
      <menuchoice>
  <shortcut>
    <keycombo action="seq">
      <keysym>C-x</keysym><keysym>C-s</keysym>
    </keycombo>
  </shortcut>
  <guimenu>Files</guimenu> <guimenuitem>Save Buffer
  as</guimenuitem>
```

```
</menuchoice>, a continuación <application>emacs</application>
          responderá con el mensaje <computeroutput>File to save
  in:~/20013/doc-tut/e2</computeroutput>, restando que usted
  escriba solamente el nombre del archivo
  (<filename>hola.txt</filename>).</para>
    </sect1>
  </chapter>
</book>
```

Capítulo 4. Usando opciones más avanzadas

Continuaremos haciendo cambios a nuestro documento con el fin de ir introduciendo características más avanzadas. El resultado de las mejoras que hagamos en este capítulo está en la la sección de nombre *Segundo listado de* `herramientas.sgml`

Sobre la *metainformación*

Además de la información propiamente dicha que brinda un documento, existen características importantes de éste, como lo son el título, los autores, la licencia de publicación o la versión, que reciben el nombre de *metainformación*.

A continuación incluiremos tal tipo de datos para nuestro en nuestro ejemplo.

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
<book lang="es">
  <bookinfo>
    <date>17 de Junio de 2001</date>
    <title><acronym>DocBook</acronym> y
    <application>emacs</application></title>
    <subtitle>Una guía básica</subtitle>
    <releaseinfo>Documentación en progreso</releaseinfo>
    <authorgroup>
      <author>
<firstname>Jaime Irving</firstname>
<surname>Dávila</surname>
      </author>
    </authorgroup>
    <address>jadavila@uniandes.edu.co</address>
    <legalnotice>
      <para>Este documento se cede al dominio público.</para>
    </legalnotice>
    <revhistory>
      <revision>
<revnumber>1.0</revnumber>
<date>16-06-2001</date>
<authorinitials>jid</authorinitials>
<revremark>Documento inicial</revremark>
      </revision>
      <revision>
<revnumber>1.1</revnumber>
<date>17-06-2001</date>
<authorinitials>jid</authorinitials>
<revremark>Inclusión encabezado</revremark>
      </revision>
      <revision>
<revnumber>1.2</revnumber>
<date>23-09-2001</date>
<authorinitials>jid</authorinitials>
<revremark>Corrección de legalnotice y jpg's</revremark>
      </revision>
```

```
      <revision>
<revnumber>1.3</revnumber>
<date>14-04-2002</date>
<revremark>Inclusión de listados de programas, faq y
marquillas de importancia</revremark>
      </revision>
  </bookinfo>
```

Todo documento que se encuentre en la red debe informar quien ejerce sus derechos de reproducción;[1] para informar el titular del derecho de reproducción de una obra se utiliza la marquilla `holder` dentro de la marquilla `copyright`, de igual forma el año (o años) de su publicación se informa a través de la marquilla `year`. Adicionalmente se deben incluir las condiciones bajo las cuáles dicho documento se puede copiar, modificar y redistribuir (usando la marquilla `legalnotice`), en este caso dicho ejemplo se cede al dominio público, esto significa que usted puede modificar o hacer con dicho documento lo que usted quiera. Hay razones importantes para ceder las obras al dominio público como son expuestas en este escrito de Vladimir Támara (http://www.tamarapatino.com/vladimir/dominio_publico.shtml).

Otra marquilla importante es el historial de revisiones (`revhistory`) en el cual a través de las marquillas de `revision` se hace explícito quien ha hecho modificaciones (`authorinitials`), en que fecha se han hecho (`date`), la versión de la revisión (`revnumber`) y una pequeña descripción de tal revisión (`revremark`).

Mostrando relaciones

Es usual que sea necesario mostrar las relaciones existentes entre las diversas partes de un documento y fuentes externas como Internet a través del uso de referencias. En esta sección mostraremos como se resuelve dicho problema usando DocBook

Referencias dentro del mismo documento

Todas las marquillas de DocBook tienen el atributo *id*, el cual permite asignarles como identificador una cadena determinada. Suele ser una buena idea asignarles identificadores a los elementos como `book`, `chapter`, `sect1` y `sect2` con el objetivo de hacer referencias a dichas divisiones del documento posteriormente. A continuación daremos identificadores a tales partes de nuestro ejemplo.

Importante: El texto del identificador debe ir entre comillas y no puede contener caracteres propios del castellano como ñ o vocales con tilde

Sugerencia: Para facilidad de navegación entre marquillas de DocBook es útil usa Move⟶Beginning of Element (**M-C-a**)y Move⟶End of Element (**M-C-e**)

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
<book lang="es" id="herramientas">
  <chapter id="DocBook">
```

```
    <title><acronym>DocBook</acronym></title>
    <para></para>
    <sect1 id="historia">
      <title>Historia</title>
      <para></para>
    </sect1>
    <sect1 id="marquillas">
      <title>Marquillas</title>
      <para></para>
    </sect1>
  </chapter>
  <chapter id="emacs">
    <title><application>emacs</application></title>
    <para></para>
    <sect1 id="invocacion">
      <title>Invocación</title>
      <para></para>
    </sect1>
    <sect1 id="persistencia">
      <title>Escribiendo y guardando un archivo</title>
    </sect1>
  </chapter>
</book>
```

La marquilla útil para hacer referencias es `xref`, simplemente en su atributo *`linkend`* se coloca el identificador del sitio a donde se hizo la referencia. A continuación presentamos un ejemplo de este tipo de situación.

```
<sect1 id="persistencia">
   <title>Escribiendo y guardando un archivo</title>

    <para>Aunque de escasa utilidad en este ámbito, pero con fines
    educativos podemos recordar <xref linkend="marquillas"/>.
    </para>
```

Esto produciría un resultado similar al siguiente

Aunque de escasa utilidad en este ámbito, pero con fines educativos podemos recordar la sección de nombre Marquillas en Capítulo 1.

En un hoja HTML dicha referencia serían adicionalmente un enlace a la sección correspondiente.

Vínculos con internet

Es usual tener que hacer vínculos con recursos que se encuentran en internet o que son archivos de un sitio local. En estos casos la marquilla adecuada es `ulink`. En caso de querer incluir una dirección de correo electrónico, se utiliza la marquilla `email`. A continuación mostramos un ejemplo que ilustra dicho tipo de situaciones.

```
<para>Puede conseguir información adicional de
<application>emacs</application> en este <ulink
 url="http://www.emacs.org">enlace</ulink>, o una copia del archivo
<ulink url="hola.txt">hola.txt</ulink>. En caso de dudas o
 comentarios puede enviar un <foreignphrase>email</foreignphrase> a
 <email>jadavila@uniandes.edu.co</email>.</para>
```

Y se ve de la siguiente forma:

Puede conseguir información adicional de emacs en este enlace (`http://www.emacs.org`), o una copia del archivo `hola.txt`. En caso de dudas o comentarios puede enviar un *email* a `<jadavila@uniandes.edu.co>`.

Capturando la atención del lector

Existe en ciertas ocasiones la necesidad de llamar la atención del lector sobre cierto tipo de información que no corresponde al *flujo* de la narración que se sigue en un documento. La forma más común de hacer esto es a través de notas a pié de página, sin embargo existen otras formas de hacerlo que presentaremos a lo largo de esta sección.

Marquillas de importancia

Dichas marquillas son usadas con el fin de resaltar o señalar cierto tipo de información que se encuentra en el documento. A continuación mostramos un ejemplo de dicha situación usando la marquilla `caution`

```
<caution>
  <para>En <application>emacs</application> <keysym>C-x</keysym>
    significa que presione al mismo tiempo
    <keycap>Control</keycap> y <keycap>X</keycap></para>
</caution>
```

Atención
En emacs C-x significa que presione al mismo tiempo **Control** y **X**

Sugerencia: Si usted desea colocar un determinado texto dentro de unas marquillas determinadas (como en el caso anterior) puede hacerlo marcando un bloque (para hacer esto último desde teclado, basta presionar C-SPC para marcar el inicio de la palabra y luego moverse al final de `para` usando M-C-e) y luego presionar C-c C-r. Posteriormente puede seleccionar que marquilla desea utilizar.

Cabe anotar que además de `caution` existen marquillas como `tip` y `important`, a continuación presentamos un ejemplo de su codificación y de como lucen en la presentación final

```
<para>Esto es simplemente un párrafo</para>
<important>
  <para>Esto es algo importante</para>
</important>
<tip>
   <para>Esto es un tip</para>
</tip>
```

Esto es simplemente un párrafo

> **Importante:** Esto es algo importante

> **Sugerencia:** Esto es un tip

Notas a pie de página

Para incluir éstas basta usar `footnote`, como lo muestra el siguiente ejemplo

```
<para><firstterm linkend="doc"><acronym>DocBook</acronym></firstterm>
  es un lenguaje de marcado, que permite escribir documentación
  técnica, nacido en 1991.
  <footnote>
    <para>En caso de querer mayor información sobre la historia,
    de <acronym>DocBook</acronym> consultar
   <ulink url="http://www.xml.com/lpt/a/1999/10/docbook/docbook-making.html"></ulink></para>
  </footnote></para>
```

que luce de la siguiente forma

DocBook es un lenguaje de marcado, que permite escribir documentación técnica, nacido en 1991. [2]

Facilitando la búsquedas al usuario

Una vez un documento adquiere un cierto tamaño, es útil dotarlo de estructuras que permitan a los usuarios consultar información sobre un tema específico o consultar la definición de una palabra que se utiliza constantemente. En esta sección trataremos el tema de la creación de índices y glosarios, que son justo la solución a dicho problema

Glosarios

Un glosario es una parte del documento en la cuál se dá la definición de algunos de los conceptos usados en él. Usualmente los glosarios se ubican al final del documento, por ello la marquilla `glossary` va usualmente después del último capítulo. A continuación presentamos un ejemplo de un glosario.

```
<glossary>
<glossentry id="doc">
<glossterm><acronym>DocBook</acronym></glossterm>
<glossdef>
<para>Lenguaje de marcado definido en
<acronym>SGML</acronym>que permite escribir documentación
técnica</para>
/glossdef>
/glossentry>
<glossentry id="em">
<glossterm>emacs</glossterm>
<glossdef>
<para>Editor de amplio uso en <acronym>unix</acronym>. Su
nombre proviene de Editor MACroS (Macros de Edición). </para>
</glossdef>
</glossentry>
</glossary>
```

Aquí básicamente declaramos declaramos un glosario usando la marquilla `glossary` y cada una de las entradas de dicho glosario se declara usando `glossentry`. Tales entradas consisten en esencia de un término (`glossterm`) y una definición (`glossdef`). Notemos que cada una de las entradas de este ejemplo tienen un identificador, esto es útil en caso de que uno quiera remitir al glosario para la definición de algún término.

Es usual denotar cuando una palabra está en un glosario o cuando es la primera ocurrencia de un término en una obra, para ello son útiles las marquillas `firstterm` y `glossterm` respectivamente. Dichas marquillas pueden contener o no un enlace a su respectiva definición, utilizándose para ello el atributo *`linkend`*. En este caso dicho atributo se hace igual al identificador correspondiente a la definición del término en el glosario. A continuación presentamos dos ejemplos de dicha situación, el primero con enlace y el segundo sin él.

```
<para><firstterm linkend="doc"><acronym>DocBook</acronym></firstterm>
 es un lenguaje de marcado, que permite escribir documentación
 técnica, nacido en 1991.</para>
```

```
<para>Para invocar a <glossterm><acronym>emacs</acronym></glossterm>,
  basta hacer lo siguiente desde una línea de comandos</para>
```

Definiendo las palabras claves de un índice

Un índice contiene un conjunto de referencias a las ocurrencias de palabras claves dentro de un documento, por ende el paso básico en la creación de un índice corresponde en señalar los sitios donde se encuentran dichas palabras a través de la marquilla `indexterm`, como lo muestra el siguiente ejemplo

```
<para>A continuación mostramos una tabla con algunas de las
   marquillas más usadas.</para>
<indexterm>
   <primary>marquillas</primary>
</indexterm>
```

Es de notar el uso de `primary` que indica que la primer palabra clave del índice es en este caso `marquillas`. Uno puede usar `secondary` en caso de querer refinar mejor la clasificación del índice como a continuación

```
...
</table>
<indexterm>
  <primary>marquilla</primary>
  <secondary>book</secondary>
</indexterm>
<indexterm>
   <primary>marquilla</primary>
   <secondary>chapter</secondary>
</indexterm>
```

> **Sugerencia:** Para generar una entrada de índice a partir de una palabra, se puede usar usar C-c **x** mientras se selecciona una palabra (para hacer esto último desde teclado, basta presionar C-SPC y luego M-C-e para ir al final de la marquilla)

Generando índices

La generación de dichos índices se describe con detalles en la página Automatic Indexing with the DocBook DSSSL Stylesheets (`http://docbook.sourceforge.net/projects/dsssl/doc/collateindex.html`). A continuación presentamos los pasos descritos en ella

1. Asegúrese de tener a `collateindex.pl`

   Es necesario que verifique que tenga dicho archivo dentro de la ruta de su sistema, para ello digite lo siguiente desde la interfaz de comandos:

   ```
   [jdavila@abadon herramientas]$ perl -S collateindex.pl
   ```

Si obtiene una descripción de la sintaxis de `collateindex.pl` podrá proseguir, en caso contrario es necesario colocar dicho archivo dentro de la ruta (en este caso haremos un enlace simbólico a `/usr/local/bin`). Para ello después de registrarse como superusuario (usando el comando **su** desde la interfaz de comandos) es necesario realizar los siguientes pasos

```
[root@abadon herramientas]$ cd /usr/share/sgml/docbook/dsssl/nwalsh-modular/bin
[root@abadon bin]$ chmod +x collateindex.pl
[root@abadon bin]$ ln -s collateindex.pl /usr/local/bin/collateindex.pl
[root@abadon bin]$ exit
[jdavila@abadon herramientas]$
```

> **Importante:** En mi caso las hojas de estilo DSSSL están ubicadas en `/usr/share/sgml/docbook/dsssl/nwalsh-modular/`, sin embargo dependiendo de su distribución pueden estar en otro lugar. Averigue en que sitio se encuentran en su distribución usando un comando como **`find / -name docbook.dsl`**

2.

Cree una referencia hacia el índice

Dentro del documento original es necesario que cree una entidad que haga referencia al archivo de índice que se va a generar. Para ello es necesario que modifique su documento que lucía de la siguiente forma

```
<!DOCTYPE Book PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
<book lang="es">
<!- Aquí viene el contenido del documento ->
...
<!- Aquí termina el contenido del documento ->
<book>
```

por algo de la siguiente forma

```
<!DOCTYPE Book PUBLIC "-//OASIS//DTD DocBook V4.1//EN"[
<!ENTITITY genindex.sgml SYSTEM "genindex.sgml">]>
<book lang="es">
<!- Aquí viene el contenido del documento ->
...
<!- Aquí termina el contenido del documento ->
&genindex.sgml
<book>
```

3.

Cree un índice vacío

Es necesario que cree un archivo `genindex.sgml` vacío mediante el siguiente comando

```
[jdavila@abadon herramientas]$ perl -S collateindex.pl -N -o genindex.sgml
```

4.

Genere un archivo `HTML.index`

El archivo `HTML.index` contiene información útil para la generación de índices, esto se puede hacer a través del siguiente comado

```
[jdavila@abadon herramientas]$ openjade -t sgml -d /usr/share/sgml/docbook/dsssl/nwalsh-modular/html/docbook.dsl -V html-index herramientas.sgml
```

5.

Genere el archivo `genindex.sgml`

Para generar el archivo de índice basta usar un comando como el siguiente

```
[jdavila@abadon herramientas]$ perl -S collateindex.pl -o genindex.sgml HTML.index
```

6.

Genere la salida deseada

Ahora puede generar la salida deseada (por ejemplo HTML o ps) cómo es descrito en la la sección de nombre *Generación de Documentos* en Capítulo 2

Dando información adicional

Una vez escrito un documento, es útil incluirle anexos en los cuales se profundiza algunos de los temas tratados en la estructura principal. A continuación presentamos dos anexos que serán parte de nuestro pequeño documento.

Incluyendo listados de programas

En nuestro pequeño documento vamos a incluir un apéndice con un listado del archivo `hola.txt`, para ello basta que usemos las marquillas `programlisting`. Es usual además que dichas listados vayan incluidas dentro de ejemplos, para lo cuál es útil usar la marquilla `example` o `informalexample`. A continuación presentamos un ejemplo usado en este documento y como es mostrado.

```
<appendix id="archivo-hola">
  <title>El archivo <filename>hola.txt</filename></title>
  <para>Por motivos de referencia incluímos el contenido del
  archivo <filename>hola.txt</filename></para>

  <informalexample>
<programlisting>
Hola Mundo
</programlisting>
  </informalexample>
```

```
</appendix>
```

```
Hola Mundo
```

> **Atención**
>
> Note que la marquilla `programlisting` está justificada a la izquierda, es necesario hacer esto pues los espacios dentro de `programlisting` son significativos.

En caso de querer incluir un listado que incluya caracteres como < o &, el contenido dentro de `programlisting` se incluye dentro de CDATA, como se muestra a continuación

```
   <informalexample>
<programlisting<![CDATA[
Hola <emphasis>Mundo</emphasis>
]]>
</programlisting>
   </informalexample>
```

```
Hola <emphasis>Mundo</emphasis>
```

Preguntas Frecuentes

La sección de preguntas frecuentes o faq (*Frequently Asked Questions*) es quizás una de las más populares y leídas dentro de un documento técnico. A continuación presentamos un ejemplo de esto, como un apéndice de `herramientas.sgml`

```
<appendix id="faq">
  <title>Preguntas Frecuentes</title>
  <qandaset defaultlabel="number">
    <qandaentry>
      <question>
        <para>¿Qué es <acronym>DocBook</acronym>?</para>
      </question>
      <answer>
        <para>Es un lenguaje de marcado útil para escribir
         documentación técnica.</para>
      </answer>
    </qandaentry>
  </qandaset>
</appendix>
```

1. ¿Qué es DocBook?

Es un lenguaje de marcado útil para escribir documentación técnica.

Sugerencia: En caso de querer cambiar la numeración de las preguntas frecuentes, se utiliza el atributo defaultlabel. Las posibles opciones para dicho atributo son number, qanda y none.

Segundo listado de `herramientas.sgml`

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
<book lang="es" id="doc-emacs">
  <bookinfo>
    <date>17 de Junio de 2001</date>
    <title><acronym>DocBook</acronym> y
    <application>emacs</application></title>
    <subtitle>Una guía básica</subtitle>
    <releaseinfo>Documentación en progreso</releaseinfo>
    <authorgroup>
      <author>
<firstname>Jaime Irving</firstname>
<surname>Dávila</surname>
      </author>
    </authorgroup>
    <address>jadavila@uniandes.edu.co</address>
    <legalnotice>
      <para>El siguiente documento se cede al dominio público</para>
    </legalnotice>
    <revhistory>
      <revision>
<revnumber>1.0</revnumber>
<date>16-06-2001</date>
<authorinitials>jid</authorinitials>
<revremark>Creación del documento inicial</revremark>
      </revision>
      <revision>
<revnumber>1.1</revnumber>
<date>17-06-2001</date>
<authorinitials>jid</authorinitials>
<revremark>Inclusión del encabezado del documento</revremark>
      </revision>
      <revision>
<revnumber>1.2</revnumber>
<date>23-09-2001</date>
<authorinitials>jid</authorinitials>
<revremark>Corrección de legalnotice y jpg's</revremark>
      </revision>
      <revision>
<revnumber>1.3</revnumber>
<date>14-04-2002</date>
<revremark>Inclusión de listados de programas, faq y
marquillas de importancia</revremark>
      </revision>
```

```
    </revhistory>
  </bookinfo>
  <chapter id="DocBook">

    <title><acronym>DocBook</acronym></title>
    <sect1 id="historia">
      <title>Historia</title>

      <para><firstterm linkend="doc"><acronym>DocBook</acronym></firstterm>
      es un lenguaje de marcado, que permite escribir documentación
      técnica, nacido en 1991.
<footnote>
  <para>En caso de querer mayor información sobre la historia,
    de <acronym>DocBook</acronym> consultar
    <ulink url="http://www.xml.com/lpt/a/1999/10/docbook/docbook-making.html"></ulink></para>
</footnote></para>

      <para>Los principales contribuyentes a dicho proyecto han
      sido:</para>
      <itemizedlist>
<listitem>
  <para>Hal Computer Systems y O’Reilly &amp; Associates, de
       1991 a 1994</para>
  <indexterm>
    <primary>contribuyente a DocBook</primary>
    <secondary>Hal Computer Systems</secondary>
  </indexterm>
  <indexterm>
    <primary>contribuyente a DocBook</primary>
    <secondary>O’Reilly &amp; Associates</secondary>
  </indexterm>
</listitem>
<listitem>
  <para>El grupo Davenport, de 1994 a 1998.</para>
  <indexterm>
    <primary>contribuyente a DocBook</primary>
    <secondary>grupo Davenport</secondary>
  </indexterm>
</listitem>
<listitem>
  <para>El grupo <acronym>OASIS</acronym> de 1998 hasta
  hoy.</para>
  <indexterm>
    <primary>contribuyente a DocBook</primary>
    <secondary>Grupo Oasis</secondary>
  </indexterm>
</listitem>
      </itemizedlist>
    </sect1>
    <sect1 id="marquillas">
      <title>Marquillas</title>

      <para>En <acronym>DocBook</acronym>, la estructura de un
```

```
      documento se delimita a través de marquillas de inicio y
      fin. Dichas marquillas lucen correspondiente como
      <replaceable>&lt;marquilla></replaceable> y
      <replaceable>&lt;\marquilla></replaceable>.</para>

      <para>A continuación mostramos una tabla con algunas de las
      marquillas más usadas</para>
      <indexterm>
<primary>marquillas</primary>
      </indexterm>

      <table>
<title>Algunas marquillas</title>
<tgroup cols="2">
  <thead>
    <row>
      <entry>Nombre de la marquilla</entry>
      <entry>Descripción de la marquilla</entry>
    </row>
  </thead>
  <tbody>
    <row>
      <entry><sgmltag>book</sgmltag></entry>
      <entry>Es la más importante, indica el inicio y fin de
      un libro</entry>
    </row>
    <row>
      <entry><sgmltag>chapter</sgmltag></entry>
      <entry>Indica el inicio y fin de un capítulo</entry>
    </row>
  </tbody>
</tgroup>
      </table>
    </sect1>
  </chapter>
  <chapter id="emacs">
    <title><application>emacs</application></title>
    <sect1 id="invocacion">
      <title>Invocación</title>

      <indexterm>
<primary>emacs</primary>
<secondary>invocación</secondary>
      </indexterm>
      <para>Para invocar a
      <glossterm><acronym>emacs</acronym></glossterm>, basta hacer lo
      siguiente desde una línea de comandos</para>

<screen><prompt>[irving@abadon e2]$ </prompt><userinput>emacs&</userinput>
<computeroutput>[1] 6251</computeroutput>
<prompt>[irving@abadon e2]$ </prompt>
</screen>
```

```
      <para>Esto produce que se abra una ventana de
      <application>emacs</application> como la siguiente:</para>

      <informalfigure>
<screenshot>
  <mediaobject>
    <imageobject>
      <imagedata fileref="emacs.eps" format="eps" scale="40">
    </imageobject>
    <imageobject>
      <imagedata fileref="emacs.png" format="png">
    </imageobject>
    <textobject>
      <phrase>Una ventana de
      <application>emacs</application></phrase>
    </textobject>
  </mediaobject>
</screenshot>
      </informalfigure>
    </sect1>
    <sect1 id="persistencia">
      <title>Escribiendo y guardando un archivo</title>
      <tip>
<para>Aunque de escasa utilidad en este ámbito, pero con fines
  educativos podemos recordar <xref linkend="marquillas"></para>
      </tip>
      <para>Dentro del editor escriba la frase <userinput>Hola
      Mundo</userinput> y posteriormente guarde dicho archivo usando
      <menuchoice>
  <shortcut>
    <keycombo action="seq">
      <keycombo>
<keysym>C-x</keysym><keysym>C-s</keysym>
      </keycombo>
    </keycombo>
  </shortcut>
  <guimenu>Files</guimenu> <guimenuitem>Save Buffer
  as</guimenuitem>
</menuchoice>, a continuación <application>emacs</application>
responderá con el mensaje <computeroutput>File to save
  in:~/20013/doc-tut/e2</computeroutput>, restando que usted
escriba solamente el nombre del archivo
(<filename>hola.txt</filename>).</para>

      <caution>
<para>En <application>emacs</application> <keysym>C-x</keysym>
  significa que presione al mismo tiempo
  <keycap>Control</keycap> y <keycap>X</keycap></para>
      </caution>

      <important>
      <para>Puede conseguir información adicional de
      <application>emacs</application> en este <ulink
```

```
      url="http://www.emacs.org">enlace</ulink>, o una copia del
      archivo <ulink url="hola.txt">hola.txt</ulink>. En caso de dudas
      o comentarios puede enviar un
      <foreignphrase>email</foreignphrase> a
      <email>jadavila@uniandes.edu.co</email>.</para></important>

      <indexterm>
<primary>hola.txt</primary>
      </indexterm>
      <indexterm>
<primary>emacs</primary>
<secondary>información</secondary>
      </indexterm>

    </sect1>
  </chapter>
  <appendix id="archivo-hola">
    <title>El archivo <filename>hola.txt</filename></title>
    <para>Por motivos de referencia incluímos el contenido del
      archivo <filename>hola.txt</filename></para>

    <informalexample>
<programlisting>
Hola Mundo
</programlisting>
    </informalexample>
  </appendix>

  <appendix id="faq">
    <title>Preguntas Frecuentes</title>
    <qandaset defaultlabel="number">
      <qandaentry>
<question>
  <para>¿Qué es <acronym>DocBook</acronym>?</para>
</question>
<answer>
  <para>Es un lenguaje de marcado útil para escribir
    documentación técnica.</para>
</answer>
      </qandaentry>
    </qandaset>
  </appendix>
  <glossary>
    <glossentry id="doc">
      <glossterm><acronym>DocBook</acronym></glossterm>
      <glossdef>
<para>Lenguaje de marcado definido en
<acronym>SGML</acronym>que permite escribir documentación
técnica</para>
      </glossdef>
    </glossentry>
    <glossentry>
      <glossterm>emacs</glossterm>
```

```
      <glossdef>
<para>Editor de amplio uso en <acronym>unix</acronym>. Su
nombre proviene de Editor MACroS (Macros de Edición). </para>
      </glossdef>
    </glossentry>
  </glossary>
</book>
```

Notas

1. Esto se hace en caso de que se desee ejercer tal tipo de derechos. Si no se desea ejercer esta clase de derechos, se dice que tal obra pertenece al dominio público y se deberá dar información explícita de esto, usando la marquilla `legalnotice`.
2. En caso de querer mayor información sobre la historia, de DocBook consultar `http://www.xml.com/lpt/a/1999/10/docbook/docbook-making.html`

Capítulo 5. Mejorando la presentación y organización del documento

A lo largo de este capítulo presentaremos maneras en las cuales podremos mejorar a nuestro documento de forma óptima en diversos aspectos, como la presentación y modularización.

Dividiendo el documento

Cuando el documento adquiere un tamaño considerable es usual querer distribuir el texto que contiene a lo largo de diversos archivos distintos. En nuestro caso dividiremos a nuestro documento en cuatro archivos distintos, `herramientas.sgml` que contendrá la *metainformación* del documento, `docbook.sgml` y `emacs.sgml` que contendrán los capítulos 1 y 2 respectivamente y `apéndices.sgml` que contendrá los apéndices.

A continuación mostramos el listado de `herramientas.sgml`

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.1//EN"[
<!ENTITY genindice.sgml SYSTEM "genindice.sgml">
<!ENTITY docbook.sgml SYSTEM "docbook.sgml">
<!ENTITY emacs.sgml SYSTEM "emacs.sgml">
<!ENTITY apendices.sgml SYSTEM "apendices.sgml">
]>
<book lang="es" id="doc-emacs">
  <bookinfo>
    <date>17 de Junio de 2001</date>
    <title><acronym>DocBook</acronym> y
    <application>emacs</application></title>
    <subtitle>Una guía básica</subtitle>
    <releaseinfo>Documentación en progreso</releaseinfo>
    <authorgroup>
      <author>
<firstname>Jaime Irving</firstname>
<surname>Dávila</surname>
      </author>
    </authorgroup>
    <address>jadavila@uniandes.edu.co</address>
    <legalnotice>
      <para>El siguiente documento se cede al dominio público</para>
    </legalnotice>
    <revhistory>
      <revision>
<revnumber>1.0</revnumber>
<date>16-06-2001</date>
<authorinitials>jid</authorinitials>
<revremark>Creación del documento inicial</revremark>
      </revision>
      <revision>
<revnumber>1.1</revnumber>
<date>17-06-2001</date>
```

```
<authorinitials>jid</authorinitials>
<revremark>Inclusión del encabezado del documento</revremark>
      </revision>
      <revision>
<revnumber>1.2</revnumber>
<date>23-09-2001</date>
<authorinitials>jid</authorinitials>
<revremark>Corrección de legalnotice y jpg's</revremark>
      </revision>
      <revision>
<revnumber>1.3</revnumber>
<date>14-04-2002</date>
<revremark>Inclusión de listados de programas, faq y
marquillas de importancia</revremark>
      </revision>
      <revision>
<revnumber>1.4</revnumber>
<date>15-04-2002</date>
<revremark>Partición en varios documentos</revremark>
      </revision>
    </revhistory>
  </bookinfo>
&docbook.sgml;
&emacs.sgml;
&apendices.sgml;
&genindex.sgml;
</book>
```

Notemos que el procedimiento es en general definir *entidades* por cada uno de los archivos a incluir a través de **`<!ENTITY ..>`** y luego incluir el contenido de dichas entidades en la estructura del documento, a través de un comando como **`&arch.sgml;`**

> **Sugerencia:** Es útil decir en los archivos que se dividió el documento (en este caso `docbook.sgml`, `emacs.sgml` y `apendices.sgml`) cuál es el documento principal al que pertenecen (en este caso `herramientas.sgml`). Esto se hace a través de un comentario al final del documento del siguiente estilo
>
> ```
> <!- Local Variables: ->
> <!- sgml-parent-document: "herramientas.sgml" ->
> <!- End: ->
> ```
>
> Incluir dichas línes permite que el modo PSGML de emacs funcione correctamente.

Automatización

Existen muchas tareas de naturaleza repetitiva a la hora de generar una salida en html o en ps a partir de un documento escrito en DocBook. Algunas de ellas incluyen la generación de índices o la copia de los archivos gráficos en los sitios indicados. Para realizar este tipo de actividades es usual utilizar un archivo llamado *Makefile* que contiene la manera como se realizan dichas actividades.

> **Importante:** Asegúrese de tener el archivo `collateindex.pl` dentro de su ruta y con permisos de ejecución, cómo es descrito en el Procedimiento 4.1,

A continuación tomaremos los archivos correspondientes a nuestro ejemplo y los dejaremos en una estructura de directorios de tal forma que podamos usar nuestro *Makefile*. Tales archivos se encuentran disponibles en el archivo `herramientas.tar.gz`

> **Sugerencia:** En caso de que desee ver cómo es la estructura final de dichos archivos, puede obtenerla en el siguiente archivo `herramientas2.tar.gz`

1. Creación del directorio de trabajo

   Cree un directorio con el nombre del documento que quiere hacer y ubíquese dentro de él, desde la interfaz de comandos esto se hace de la siguiente forma:

   ```
   [jdavila@abadon jdavila]$ mkdir herramientas
   [jdavila@abadon jdavila]$ cd herramientas
   ```

2. Creación de subdirectorios

   Dentro de este directorio cree los sudirectorios `src`, `png` y `others`, de la siguiente forma

   ```
   [jdavila@abadon herramientas]$ mkdir src
   [jdavila@abadon herramientas]$ mkdir png
   [jdavila@abadon herramientas]$ mkdir others
   ```

3. Descomprima el contenido del archivo `herramientas.tar.gz`

   Para ello basta con que digite desde la interfaz de comandos (tenga en cuenta que estamos suponiendo que `herramientas.tar.gz` está dentro del directorio `herramientas`)

   ```
   [jdavila@abadon herramientas]$ tar xvfz herramientas.tar.gz
   ```

4. Copia y edición del *Makefile*

   Copie el archivo `Makefile` en su directorio `herramientas` con el nombre `Makefile` y a continuación edítelo cambiando las líneas

   ```
   DOC=dbktut
   INDEX=genindex
   ```

, por la líneas

```
DOC=herramientas
INDEX=genindice
```

En general dichas líneas deben ser cambiadas por el nombre del archivo que usted escribió y por el nombre del archivo índice (sin la extensión .sgml)

> **Importante:** Dentro del archivo `Makefile` se encuentran definidas diversas variables que permiten configurar las diversas salidas que este produce. Tenga en cuenta que para generar la salida html es necesario que tenga definidas adecuadamente las variables SINGLEDB-STYLESHEET y DBSTYLESHEET, dichas variables especifican el sitio donde se encuentran las hojas de estilo para la salida html. En nuestro caso estamos usando la hoja de estilo ldp.dsl (`http://www.tldp.org/authors/tools/ldp.dsl`) para la salida html.

5.

Copie los archivos fuente

Copie los archivos con extensión sgml en el directorio `src` con un comando como el siguiente

```
[jdavila@abadon herramientas]$ mv *.sgml src
```

6.

Copie los gráficos

Copie los archivos gráficos en formato png en el directorio `png`, usando de la siguiente forma la interfaz de comandos

```
[jdavila@abadon herramientas]$ mv *.png png
```

7.

Copie los demás archivos

A veces es útil tener archivo anexos con el documento que no son las gráficas (en nuestro ejemplo el archivo `hola.txt`).Copie tales archivos en el directorio `others`, mediante la siguiente orden

```
[jdavila@abadon herramientas]$ mv hola.txt others
```

8.

Use **Make**

Ahora basta con que, ubicado desde la interfaz de comandos en el directorio `herramientas` escriba

```
[jdavila@abadon herramientas]$ make
```

Esto generará un conjunto de mensajes en su interfaz de comandos, que indican que se están creando los documentos html, ps y pdf a partir de su documento.

9.

Vea los archivos generados

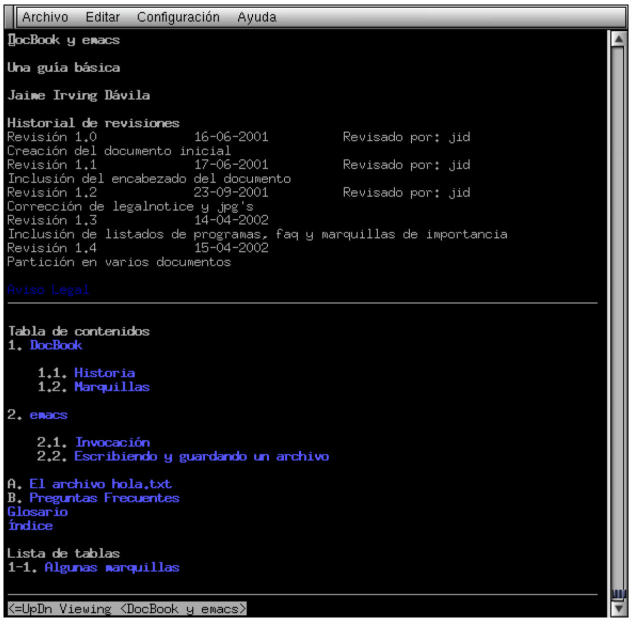
Una vez ejecutado el comando **`make`** se generan un conjunto de archivos html, pdf y ps. Los archivos html están ubicados en los directorios `out-html` y `out-htmls`; en el primer directorio está la salida html como un sólo archivo, mientras en el segundo directorio está dicha salida como varios archivos html

Si desea ver los archivos generados basta que abre en su navegador de preferencia, el archivo de nombre `herramientas.html` que se encuentra ubicado en el directorio `out-html`

En caso de que desee abrir un navegador que solo sirva en modo texto le recomendamos usar w3m (`http://w3m.sourceforge.net`), para ello basta hacer lo siguiente desde la interfaz de comandos

```
[jdavila@abadon herramientas]$ w3m out-htmls/index.html
```

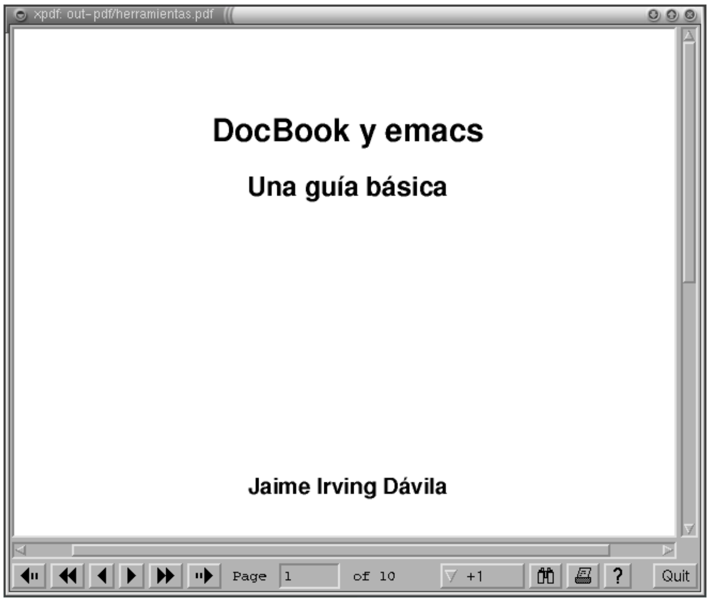
y obtendrá un resultado como el siguiente:



En caso de que desee ver el archivo pdf generado basta que use xpdf (`http://www.foolabs.com/xpdf`)

```
[jdavila@abadon herramientas]$ xpdf out-pdf/herramientas.pdf&
```

y obtendrá



10.

Conozca otras opciones del **make**

Como habrá notado la generación de los documentos html, ps y pdf demora unos cuantos segundos (más en el caso de ps o del pdf). Por lo tanto es útil tener la opción de generar solamente un sólo html, varios html o el documento ps, para ello basta usar la opción

```
[jdavila@abadon herramientas]$ make single-html
```

En caso de querer generar la salida en un sólo html o la opción:

```
[jdavila@abadon herramientas]$ make multiple-html
```

En caso de querer generar varios html. Para generar la salida ps, basta:

```
[jdavila@abadon herramientas]$ make ps
```

En caso de querer generar su documento pdf basta usar

```
[jdavila@abadon herramientas]$ make pdf
```

En caso de que desee limpiar la estructura de su documento dejando solamente las fuentes y archivos esenciales basta con que use el comando

```
[jdavila@abadon herramientas]$ make clean
```

11.

Hágalo todo desde emacs

Para ello basta que desde emacs haga uso de **`M-!`** y luego digite **`cd ..;make`**. En caso de querer generar un sólo html reemplace **`make`** por **`make single-html`** y lo equivalente para generar ps y pdf

Acerca del estilo

Hasta el momento no nos hemos preocupado por la presentación de nuestro documento, solamente por el significado de las palabras en cada contexto. Sin embargo en algunas ocasiones uno desea formatear la salida impresa o en html de una determinada forma y en este momento es útil usar una hoja de estilo personalizada. Si quiere mayor información sobre este tipo de procedimiento lo remitimos a la página Customizing your stylesheet (`http://docbook.sourceforge.net/projects/dsssl/doc/custom.html`)

La idea de la hoja de estilo es tener un archivo en el cual uno configura ciertos parámetros que afectan las diferentes salidas generadas a partir del archivo sgml. Supongamos que la salida html satisface las siguientes condiciones

- No queremos incluir una lista de las tablas del documento.
- Queremos utilizar *fast-forwarding*, una característica que nos permite navegar fácilmente de capítulo en capítulo.
- Las páginas htmlque se generen queremos que tengan como nombre el id de cada capítulo y queremos que su extensión sea .html

En cuanto a la salida impresa, simplemente deseamos especificar las márgenes izquierda y derecha. A partir de dichas características creamos una hoja de estilo personalizada, a la que llamaremos `custom.dsl` y la cual presentamos a continuación.

```
<!DOCTYPE style-sheet PUBLIC
          "-//James Clark//DTD DSSSL Style Sheet//EN" [
<!ENTITY % html "IGNORE">
<![%html;[
<!ENTITY % print "IGNORE">
<!ENTITY docbook.dsl PUBLIC
         "-//Norman Walsh//DOCUMENT DocBook HTML Stylesheet//EN"
 CDATA dsssl>
]]>
<!ENTITY % print "INCLUDE">
<![%print;[
<!ENTITY docbook.dsl PUBLIC
         "-//Norman Walsh//DOCUMENT DocBook Print Stylesheet//EN"
 CDATA dsssl>
]]>
]>
<style-sheet>
<style-specification id="print" use="docbook">
<style-specification-body>

<!- print stylesheet ->
(define %left-margin%
  ;; Width of left margin
  4pi)

(define %left-margin%
  ;; Width of left margin
  4pi)
</style-specification-body>
```

```
</style-specification>

<style-specification id="html" use="docbook">
<style-specification-body>

<!- html stylesheet->
(define ($generate-book-lot-list$)
  ;; Which Lists of Titles should be produced for Books?
  (list (normalize "equation")))

(define %gentext-nav-use-ff%
  ;; Add "fast-forward" to the navigation links?
  #t)
(define %html-ext%
  ;; when producing HTML files, use this extension
  ".html")
(define %use-id-as-filename%
  ;; Use ID attributes as name for component HTML files?
  #t)
</style-specification-body>
</style-specification>
<external-specification id="docbook" document="docbook.dsl">
</style-sheet>
```

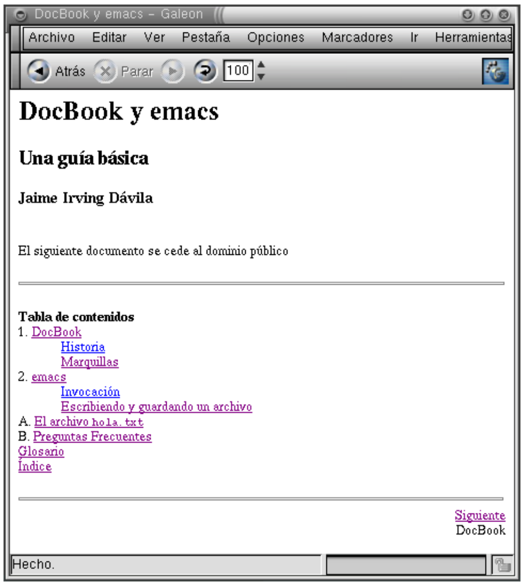
A continuación modificaremos nuestro *Makefile* para que la salida html se genere con la configuración que hemos pedido

```
#Location of the stylesheet for the html output
SINGLEDBSTYLESHEET = ldp.dsl\#html
DBSTYLESHEET = ldp.dsl\#html
```

por las líneas,

```
#Location of the stylesheet for the html output
SINGLEDBSTYLESHEET = custom.dsl\#html
DBSTYLESHEET = custom.dsl\#html
```

Si usamos el comando **`make multiple-html`**, y luego usamos galeon (`http://galeon.sourceforge.net`) para vizualizar el html tendremos que luce de la siguiente forma.



Sugerencia: Si quiere generar la salida html sin necesidad de usar el *Makefile* basta que emplee un comando como el siguiente

```
openjade -t sgml -ihtml -d custom.dsl#html herramientas.sgml
```

Usando **docbook2html** basta que emplee un comando como el siguiente:

```
docbook2html -d custom.dsl#html herramientas.sgml
```

Para que la salida impresa utilice la hoja de estilo personalizada, es necesario que las variable `SCRIPTPS` y `SCRIPTPDF` dentro del `Makefile` sean iguales a **docbook2ps -d custom.dsl#html** o a **docbook2pdf -d custom.dsl#html** (en algunos casos esto no sirve y es necesario editar los archivo `db2dvi` y `db2pdf`).

Importante: Para una lista de los parámetros configurables en la salida impresa (ps, pdf) o la salida html recomendamos la lectura de The Modular DocBook Stylesheets (`http://docbook.sourceforge.net/projects/dsssl/doc/index.html`), en especial las secciones DocBook Print Parameters (`http://docbook.sourceforge.net/projects/dsssl/doc/print/index.html`) y DocBook HTML Parameters (`http://docbook.sourceforge.net/projects/dsssl/doc/html/index.html`)

¿Donde ir por más?

Existen diversas oportunidades en las cuáles uno desea saber si existe una determinada marquilla con una cierto significado. A continuación presentamos una serie de pasos que pueden ayudar en esta labor.

1.

   ¿Es una característica semántica?

Esta es quizás la primera pregunta que uno tiene que hacerse puesto que DocBook es un lenguaje de marcado semántico o de contenido. Algunas características de contenido recurrentes son por ejemplo denotar que una palabra determinada representa una aplicación, un archivo o un comando. Características de presentación son por ejemplo el tipo de letra o el espaciamiento, tal tipo de características se controlan usando una hoja de estilo personalizada como es descrito en la la sección de nombre *Acerca del estilo*.

2.

¿Se encuentra en el listado de marquillas?

Para ello es muy útil tener a mano una copia del libro DocBook: The Definitive Guide (`http://docbook.org/tdg/en/html/docbook.html`) y consultar el apéndice, en el cuál se describen con lujo de detalles todas las posibles marquillas. DocBook tiene marquillas para entradas bibliográficas, clases, métodos, descripción de parámetros de comandos y un gran número de posibilidades.

3.

Usar la marquilla `systemitem`

A pesar de que existan cerca de 400 marquillas en DocBook, a veces no existe una marquilla adecuada, como por ejemplo si uno quiere hacer referencia a un sistema operacional. En tales casos se usa la marquilla `systemitem` con un atributo `role` apropiado. Un ejemplo de dicha situación es el siguiente **`<systemitem role="os">Windows</systemitem>`**.

De igual forma existen en DocBook los medios de comunicación usuales en la comunidad del software libre y de fuentes abiertas para resolver dudas como lo son

- Una lista de correo en castellano, docbook-ayuda (`http://listas.hispalinux.es/mailman/listinfo/docbook-ayuda`)
- Una faq (*Frequently Asked Questions*) disponible en `http://www.dpawson.co.uk/docbook/`
- Listas de correo en inglés, como lo son docbook (en ella se responden preguntas referentes al lenguaje de marcado como tal) y docbook-apps (en ella se habla de temas como presentación y herramientas que apoyan el proceso de DocBook). Para más información ver `http://www.oasis-open.org/docbook/mailinglist/index.shtml`.
- Una aplicación que le permite enviar su documento escrito en formato DocBook por correo electrónico y luego le permite obtener dicho documento en formato html o en texto plano. Para las instrucciones de su uso consulte la página DocBook Processor (`http://www.lupercalia.net/scoop/special/dbp`)
- Una página Wiki-Wiki (`http://docbook.org/wiki/`), donde se puede encontrar mucha información sobre DocBook y a su vez se puede editar o corregir de forma fácil.
- En caso de que desee una charla en tiempo real, puede entrar al canal #docbook en irc.openprojects.net

Índice